



**Universidad Nacional Autónoma de México**

**FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN**

**IMPLEMENTACIÓN DE UN  
SISTEMA DE ULTRASONIDO  
PARA LA DETECCIÓN  
DE OBJETOS**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERO MECÁNICO ELECTRICISTA**  
P R E S E N T A N :  
**HÉCTOR ALFREDO ÁLVAREZ DE LA ROSA**  
**VÍCTOR HUGO GARCÍA PICHARDO**

ASESORES:  
**DRA. LUCÍA MEDINA GÓMEZ**  
**DR. JESÚS SAVAGE CARMONA**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN  
 UNIDAD DE LA ADMINISTRACION ESCOLAR  
 DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

SECRETARIA DE EDUCACION PUBLICA  
 DEPARTAMENTO DE EXAMENES PROFESIONALES  
 CUAUTITLAN

DR. JUAN ANTONIO MONTARAZ CRESPO  
 DIRECTOR DE LA FES CUAUTITLAN  
 P R E S E N T E

ATN: Q. Ma. del Carmen García Mijares  
 Jefe del Departamento de Exámenes  
 Profesionales de la FES Cuautitlán

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicarle a usted que revisamos la TESIS:

"Implementación de un sistema de ultrasonido para la detección de objetos".

que presenta el pasante: Héctor Alfredo Alvarez de la Rosa  
 con número de cuenta: 9236249-7 para obtener el título de  
 Ingeniero Mecánico Electricista

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

ATENTAMENTE  
 "POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 24 de Febrero de 2004

PRESIDENTE	<u>Ing. Casildo Rodríguez Arciniega</u>	
VOCAL	<u>MAI. Pedro Guzmán Tinajero</u>	
SECRETARIO	<u>Dra. Lucía Medina Gómez</u>	
PRIMER SUPLENTE	<u>Ing. Roberto Reyes Arce</u>	
SEGUNDO SUPLENTE	<u>Ing. Fernando Patlán Cardoso</u>	



ESTADOS UNIDOS MEXICANOS

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

FECHA: 24 FEBRERO 2004

DR. JUAN ANTONIO MONTARAZ CRESPO
DIRECTOR DE LA FES CUAUTITLAN
PRESENTE

ATN: Q. Ma. del Carmen Garcia Mijares
Jefe del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS:

"Implementación de un sistema de ultrasonido para la detección de objetos".

que presenta el pasante: Víctor Hugo García Pichardo
con número de cuenta: 92373277 para obtener el título de
Ingeniero Mecánico Electricista

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO

ATENTAMENTE

"POR MI RAZA HABLARA EL ESPIRITU"

Cuautitlán Izcalli, Méx. a 24 de Febrero de 2004

- PRESIDENTE Ing. Casildo Rodríguez Arciniega
VOCAL MAI. Pedro Guzmán Tinajero
SECRETARIO Dra. Lucía Medina Gómez
PRIMER SUPLENTE Ing. Roberto Reyes Arce
SEGUNDO SUPLENTE Ing. Fernando Patlán Cardoso

## **Agradecimientos**

Esta tesis representa un parte final entre una etapa muy enriquecedora en el camino que el tiempo obliga. En toda la experiencia universitaria y la conclusión del trabajo de tesis, ha habido personas que merecen las gracias porque sin su valiosa aportación no hubiera sido posible este trabajo y también a quienes las merecen por haber plasmado su huella en mi camino.

A mis padres, Cristina De la Rosa González y Leonardo Álvarez Santiago y mis hermanas y hermanos, les agradezco su apoyo, su guía y su confianza en la realización de mis sueños. Soy afortunado por contar siempre con su amor, comprensión y ejemplo. Esta tesis es suya.

Una especial muestra de gratitud a Víctor García Pichardo por su amistad y trabajo, con quien realice en conjunto esta tesis, sin su colaboración la mitad de la tesis no se hubiera realizado.

A los asesores de tesis por haber creído desde el principio en mi capacidad para abordar este trabajo. Gracias Dra. Lucia Medina Gómez y Dr. Jesús Savage Carmona.

Sin lugar a duda este trabajo no pudo haberse realizado sin la formación que recibí durante cuatro años en la Facultad de Estudios Superiores Cuautitlan ( U.N.A.M.). Gracias a todos los maestros que contribuyeron realmente a mi formación.

Al Departamento de Posgrado de la Facultad de Ingeniería, y muy especial a mis compañeros de Biorobotica por haber permitido trabajar y compartir su amistad.

Profundamente al Instituto de Investigación de Matemáticas Aplicadas a Sistemas y todo su personal por las facilidades brindadas para la realización de esta tesis .

Valoro la notable colaboración del Departamento de Informática de la Facultad de Química ( U.N.A.M.), que me apoyaron y brindaron los recursos para que esta tesis llegara a buen termino.

A todos mis amigos y amigas con los que tengo el privilegio de compartir la vida, gracias.

Por último quiero dar las gracias a todas aquellas personas que con su actitud personal nos hacen la vida más llevadera, a todos aquellos que me han devuelto una sonrisa, a todos aquellos que han puesto de su parte para que el trajín diario sea más llevadero

## **Agradecimientos.**

Agradezco a Dios por toda su bondad y amor que siempre han sido la fuente de mi fe.

Dedico esta tesis a mi mamá Bertha Pichardo quien es todo un ejemplo a seguir, que durante toda mi vida me ha dado su amor y apoyo incondicional, y es debido a la admiración que siento por ella que puedo afrontar cualquier dificultad que en la vida pueda encontrar, gracias a su guía pude culminar esta parte angular en mi vida profesional.

A Nancy y a Daniel mis hermanos, que me han dado todo su apoyo y cariño que siempre estuvieron conmigo en todos los momentos importantes durante la carrera.

A mis abuelitos por estar siempre conmigo y darme su cariño.

A mis primos y tíos por darme su apoyo y amistad.

A mi compañero Héctor Álvarez, quien durante toda la carrera me dio su amistad incondicional que fue un gran apoyo durante toda la carrera y que gracias a su ayuda este proyecto pude verse terminado satisfactoriamente.

A mis asesores Dra. Lucía Medina (IIMAS) y Dr. Jesús Savage (DEPFI) por haber creído en nosotros al darnos su confianza, su apoyo y su amistad durante la realización de este trabajo, que gracias a ellos he podido crecer en lo profesional y les estoy muy agradecido. También quiero agradecer al Departamento de Posgrado de la Facultad de Ingeniería en especial al laboratorio de Biorobotica por darnos las facilidades de la realización de este proyecto así como su amistad y su apoyo.

A mis profesores de la Facultad de Estudios Superiores Cuautitlán que gracias a sus consejos y ejemplo he podido crecer profesionalmente y tener siempre el espíritu de superación y excelencia que todo profesional debe cultivar.

A la UNAM por haberme recibido con los brazos abiertos y haberme dado todas las facilidades para crecer como persona y en lo profesional.

Finalmente a todos mis amigos y amigas que me permitieron contar con el privilegio de su amistad y compartir momentos llenos de felicidad.



## ÍNDICE

<b>CAPÍTULO I INTRODUCCIÓN</b> .....	<b>1</b>
1.1 Objetivos Específicos.....	1
1.2 Metas.....	1
1.3 Justificación.....	1
1.4 Sistemas Ultrasónicos.....	2
1.5 Sistema de detección de obstáculos.....	5
1.6 Panorama de la Tesis .....	6
<b>CAPÍTULO II TARJETA NI-DAQ</b> .....	<b>8</b>
2.1 Introducción Tarjeta de Adquisición de Datos.....	8
2.1.1 Componentes de la tarjeta de adquisición de datos .....	9
2.10 Configuración .....	22
2.10.1 Funciones analógicas de Entrada y Salida.....	24
2.12 Programa de prueba .....	24
2.2 Identificación de las terminales del dispositivo SCB-68 .....	11
2.2.1 Conexiones del dispositivo SCB-68 .....	11
2.3 Instalación de los dispositivos.....	12
2.4 Instalación de controladores de la tarjeta NI-DAQ .....	13
2.5 Configurar Microsoft Visual C++, .....	16
2.6 Sistema Digital .....	16
2.6.1 Conexiones de las entradas digitales.....	17
2.6.2 Configuración .....	17
2.6.3 Funciones digitales de salida.....	17
2.6.4 Funciones digitales de Entrada/Salida.....	18
2.6.5 Programa de prueba .....	18
2.7 Sistema Analógico .....	19
2.7.1 Conexiones .....	19
2.7.2 Consideración de las conexiones .....	19
2.8 Polaridad y Rango de Entrada.....	21

---



---

2.9 Las Conexiones para el Señal Flotante (la Configuración de RSE) -----	22
2.9.1 Conexiones de Alimentación-----	22
<b>CAPÍTULO III Sonares Ultrasónicos -----</b>	<b>26</b>
3.1 Ultrasonido. -----	26
3.2 Propiedades de las ondas ultrasónicas.-----	27
3.3 Fenómenos de las señales ultrasónicas-----	28
3.3.1 Reflexión y refracción de ondas -----	28
3.4 Transductores. -----	29
3.4.1 Características Transductores. -----	29
3.5 Sonares Ultrasónicos.-----	30
3.5.1 Medición de distancia -----	30
3.6 Unidad de Medición Polaroid 6500 -----	31
3.6.1 Descripción de las Señales.-----	34
3.7 Arreglo de sonares. -----	37
<b>CAPÍTULO IV DESARROLLO-----</b>	<b>39</b>
4. 3 Conexiones entre los cables de control de los sonares-----	45
4.1 Introducción-----	39
4.10 Portabilidad del Programa-----	55
4.2 Sonares Polaroid -----	39
4.2.1 Pruebas de funcionamiento de sonares ultrasónicos-----	39
4.2.2 Arreglo de sonares-----	40
4.2.3 Modificaciones a la unidad de medición Polaroid #6500-----	42
4.4 Acoplamiento del arreglo de sonares y la tarjeta DaqCard -----	46
4.5 Activación de sonares -----	48
4.6 Configuración Individual -----	48
4.7 Configuración Serie-----	50
4.8 Captura de ecos-----	51
4.9 Pruebas -----	53

---

---

<b>CAPÍTULO V PRUEBAS Y RESULTADOS</b> .....	<b>57</b>
5.1 Introducción.....	57
5.2 Configuración de las pruebas.....	57
5.2 Configuración Individual caso I.....	58
5.2.1 Configuración Individual caso II. ....	59
5.2.2 Configuración Individual caso III. ....	61
5.3 Configuración serie caso I. ....	62
5.3.2 Configuración Serie caso II .....	63
5.3.3 Configuración Serie caso III .....	64
5.4 Conclusiones.....	65
<b>CAPÍTULO VI CONCLUSIONES Y TRABAJO A FUTURO</b> .....	<b>66</b>
6.1 Conclusiones.....	66
6.2 Trabajo a futuro y recomendaciones .....	68
<b>Apéndice A</b> .....	<b>70</b>
<b>Apéndice B</b> .....	<b>73</b>
<b>Bibliografía</b> .....	<b>86</b>

## **CAPÍTULO I INTRODUCCIÓN**

El objetivo general de este proyecto, es el desarrollo y la implementación de un sistema de detección de objetos utilizando un arreglo lineal de sonares.

### **1.1 Objetivos Específicos.**

- Implementar el sistema por medio de un arreglo de sonares.
- Desarrollar las modificaciones pertinentes al arreglo para que tenga dos tipos de configuraciones de envío y recepción de señal.
- Controlar el sistema por medio de una computadora tipo laptop.
- Utilizar una tarjeta adquisidora de datos NIDAQ para controlar el envío y recepción de las señales adecuadas.
- Visualizar las señales adquiridas en una computadora tipo laptop.
- Realizar las pruebas de funcionamiento del sistema

### **1.2 Metas**

La meta de este proyecto está definido por los objetivos generales y específicos en los que se basa la realización de este trabajo. El proyecto está enfocado en el desarrollo y la implementación de un sistema de detección y localización de objetos, por medio de un arreglo de sonares controlado por una laptop usando una tarjeta adquisidora de datos. Aunque el alcance de este proyecto es claro, el mismo proyecto es campo fértil para nuevas investigaciones sobre el área de las señales y sistemas de navegación para robots móviles.

### **1.3 Justificación.**

Para que un sistema de detección de obstáculos pueda tener una visión clara del entorno que lo rodea, es necesario el uso de arreglos de sonares y así obtener información de diferentes partes del objeto y su identificación así como reconstrucción. Si se quiere reconstruir un objeto o identificarlo es necesario obtener una gran cantidad de muestras que sólo pueden ser obtenidas por medio del arreglo de mas de un sonar, debido a que información valiosa puede ser punto ciego para un solo sonar, mientras que con el uso del arreglo de sonares se pueden obtener mas información sobre el objeto detectado.

#### **1.4 Sistemas Ultrasónicos.**

Los sistemas ultrasónicos han sido utilizados para detección, evasión y navegación de robots [1], en pruebas no destructivas [2], diagnóstico clínico y tratamientos médicos [3]. Estos sistemas consisten en un transductor o arreglos de transductores capaces de emitir un pulso ultrasónico u onda continua a cierta frecuencia de resonancia.

Los sistemas ultrasónicos pulso-eco, permiten conocer la distancia radial entre los transductores y el objeto que causa que la energía ultrasónica sea rebotada al transductor.

Cuando únicamente un transductor es utilizado para detectar y localizar fuentes reflectoras presente en la región visible del transductor, es necesario mover el transductor con una trayectoria conocida o bien utilizar un arreglo de transductores que mediante algún procesamiento digital de las señales recibidas se pueda determinar la posición y la forma de la fuente reflectora.

Los robots móviles para su sistema de navegación necesitan un transductor que pueda detectar objetos en tiempo real es aquí donde entran los sistemas ultrasónicos, los sonares permiten detectar los objetos que se van encontrando en su paso, existen arreglos en anillo donde se usan por lo regular más de 20 sonares, donde se tiene una visión de 360° cuando un sonar POLAROID [4] tiene una visión real de 30°. No solo existen los sistemas de navegación para robots móviles si no también para sistemas de evasión para personas invidentes [7].

Por conveniencia la gran mayoría de los sistemas de evasión de obstáculos emplean el sistema POLAROID [4], la diferencia entre los diferentes sistemas de ultrasonido para la evasión de obstáculos son básicamente dos; la manera en la que los sonares son acomodados y el algoritmo empleado en el procesamiento de las señales para la reconstrucción del objeto. Un método muy usado es lo que se conoce como localización absoluta [8] se basa en la construcción de segmentos (por medio de mediciones) que son comparados con el ambiente y una construcción de una cuadrícula.

La construcción de segmentos se lleva a cabo por medio de sensores ultrasónicos. Un segmento se construye con tres puntos. Un nuevo punto es asociado a un segmento si la distancia entre el segmento y el punto es menor que otro punto predefinido. Los segmentos son comparados con el ambiente al encontrar la posición del robot. Así puede construir un mapa del ambiente donde el robot se desenvuelve.

La mayoría de los sensores de ultrasonido de bajo costo se basan en la emisión de un pulso de ultrasonido, midiendo el tiempo que transcurre entre la emisión del sonido y la detección del eco, de esta forma se puede establecer la distancia radial a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora, mediante la fórmula:

$$d = \left( \frac{1}{2} \right) vt$$

donde  $v$  es la velocidad del sonido en el aire y  $t$  es el tiempo transcurrido entre la emisión y recepción del pulso. Sin embargo, factores inherentes tanto a los ultrasonidos como al mundo real, influyen de una forma determinante en las medidas realizadas. Por tanto, es necesario un conocimiento de las diversas fuentes de incertidumbre que afectan a las medidas para poder tratarlas de forma adecuada, minimizando su efecto en el conocimiento del entorno que se desea adquirir. Entre los diversos factores que las alteran cabe destacar:

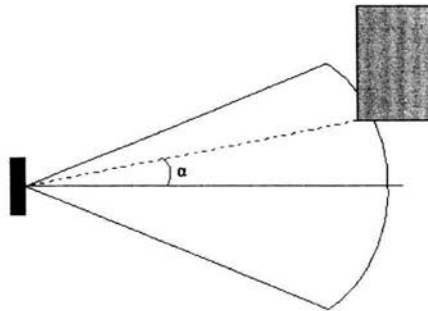


Figura 1.1 Incertidumbre angular en la medida de un ultrasonido

El campo de actuación del pulso que se emite desde un transductor de ultrasonido tiene forma cónica. El eco que se recibe como respuesta a la reflexión del sonido indica la presencia del objeto que se encuentra dentro del cono acústico y no especifica en ningún momento la localización angular del mismo.

- La cantidad de energía acústica reflejada por el obstáculo depende en gran medida de la estructura de su superficie. Para obtener una reflexión altamente difusa del obstáculo, el tamaño de las irregularidades sobre la superficie reflectora debe ser comparable a la longitud de onda de la onda de ultrasonido incidente.
- En la mayoría de sensores de ultrasonido se utiliza el mismo transductor como emisor y receptor. Tras la emisión del ultrasonido se espera un determinado tiempo a que las vibraciones en el sensor desaparezcan y esté preparado para recibir el eco producido por el obstáculo.

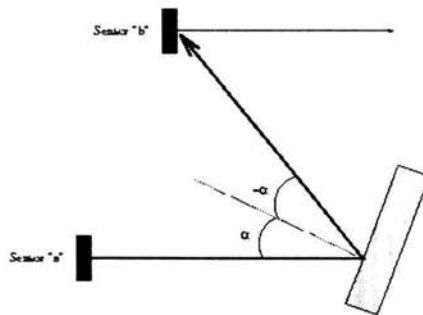


Figura 1.2 Crosstalk: El sensor "a" emite el pulso que recibe el sensor "b"

- Un factor de error muy común es el conocido como falsos ecos. Estos falsos ecos se pueden producir por razones diferentes. Una fuente común de falsos ecos, conocida como crosstalk, se produce cuando se emplea un cinturón de ultrasonidos donde una serie de sensores están trabajando al mismo tiempo, Figura 1.2.

En este caso puede ocurrir (y ocurre con una frecuencia relativamente alta) que un sensor emita un pulso y sea recibido por otro sensor que estuviese esperando el eco del pulso que él había enviado con anterioridad (o viceversa).

- Para emitir un pulso de ultrasonido hay que excitar la membrana del transductor con una señal en forma de delta de Dirac. Sin embargo, en los sensores, la señal excitadora es en la práctica un pulso cuadrado por lo que el efecto resultante es el de la emisión de todo un tren de ondas de ultrasonidos que emergen de la membrana del sensor.

### **1.5 Sistema de detección de obstáculos.**

Las partes que conforman un sistema ultrasónico para la evasión, detección y navegación de un robot, es el sistema de control, una tarjeta para emitir el pulso ultrasónico y transductores.

En el sistema de control por lo regular se utilizan microprocesadores motorola 68HC11 [9], o pics [9] el único problema es que no manejan adecuadamente el ancho de banda para un arreglo de sonares, solo funcionan eficientemente con un sonar, en la mayoría de los sistemas se utilizan tarjetas diseñadas para la adquisición de datos como la NIDAQ [5], debido a su facilidad para controlar mas de un sonar en un tiempo dado.

La tarjeta para emitir el pulso ultrasónico consiste en uno o dos circuitos integrados y una serie de transistores acoplados para que sean capaces de emitir el pulso necesario para activar el transductor y procesar la señal recibida. Por conveniencia se utiliza la tarjeta POLAROID [4] ya que es de bajo costo y tiene todas las características deseadas para procesar el eco recibido.

Los transductores empleados para los sistemas ultrasónicos tienen que comportarse como transmisor y receptor a sus respectivos tiempo, existen diferentes tipo de transductores que se explican en los siguientes capítulos. Para el proyecto se emplearon los que vienen con la tarjeta de medición POLAROID [4].



## **1.6 Panorama de la Tesis**

La tesis esta conformada por seis capítulos.

### Capítulo I

Se da un panorama de lo que se trata el proyecto y una breve introducción sobre, ultrasonido y los sistemas de navegación, básicamente se da una perspectiva del proyecto que se desarrolló.

### Capítulo II

En este apartado se describe el funcionamiento de la tarjeta adquisidora de datos NIDAQ [5], las características principales y se hace énfasis en las características empleadas en el proyecto y sus ventajas.

### Capítulo III

Aquí se habla sobre el ultrasonido, los diferentes tipos de transductores y en especial el transductor POLAROID [4], así como arreglo lineal de sonares empleado. Se proporcionan las características principales de la tarjeta de medición POLAROID [4], frecuencia, modo de operación, señales, rango visible y modo de configuración. Se explica en forma detallada el funcionamiento de los principales conectores, sus señales asociadas y como procesa el eco recibido para calcular la distancia y la señal que se utilizó para el sistema de sonares.

### Capítulo IV

Se plantea el desarrollo y la implementación del sistema de sonares, las pruebas realizadas con la tarjeta de adquisición de datos y las pruebas con la tarjeta de medición. Se proporciona en detalle la integración de la tarjeta de adquisición de datos con el arreglo lineal de sonares, el modo de operación y el control por medio de una computadora tipo laptop y el procesamiento de las señales mediante MATLAB. Los diagramas de las conexiones que se hicieron y las modificaciones que se le hicieron a la tarjeta POLAROID [4], con el fin de forzar a los transductores a funcionar con dos tipos de configuraciones por medio de un switch.

Capítulo V

Se plantean las pruebas realizadas una vez que se desarrolló el sistema ultrasónico, hacia un objeto a diferentes distancias y diferentes ángulos, para comparar las señales obtenidas en una computadora.

Capítulo VI

En este capítulo se escriben las conclusiones encontradas durante el desarrollo del sistema implementado, se dan recomendaciones para futuras modificaciones al sistema así como trabajo a futuro.

## CAPÍTULO II TARJETA NI-DAQ

### 2.1 Introducción Tarjeta de Adquisición de Datos

Se entiende por Adquisición de datos a la acción de medir variables, convertirlas a formato digital, almacenarlas en un computador y procesarlas. Este proceso necesita de una "interface" entre el mundo físico y la computadora que se suele denominar tarjeta de adquisición de datos.

El proceso de adquisición de datos del mundo físico conlleva los siguientes pasos fundamentales:

1. Utilización de un sonar/transductor que permite detectar y convertir la variable física a una señal analógica de voltaje o corriente eléctrica.
2. Amplificación de la señal de voltaje o corriente, si se requiere. Si la señal que proviene del sonar es débil, se requiere un amplificador de voltaje y algún método para filtrar los ruidos eléctricos.
3. Traducción de esta señal analógica al lenguaje propio del computador: lenguaje digital. Este proceso se conoce técnicamente como conversión ANÁLOGO/DIGITAL (A/D).
4. Adquisición propiamente dicha de los datos que, en forma digital, podrán ser almacenados en la memoria del micro y desplegado en pantalla o a otro periférico de la computadora.

La figura 2.1 muestra un diagrama conceptual del proceso de adquisición de datos, el cual sintetiza los pasos fundamentales ya mencionados.

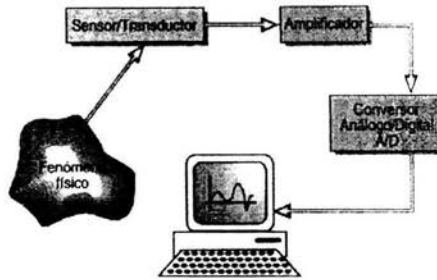


Figura 2.1 Adquisición de datos

La tarjeta DAQCard (Data Acquisition Card), es de multifunción analógica, digital, y registro de tiempo de Entrada/Salida para computadoras equipadas con ranuras de PCMCIA (Personal Computer Memory Card International Association). Esta tarjeta tiene como característica un convertidor analógico-digital de 12-bit y 16-bit con ocho líneas de Entrada/Salida digital TTL-compatible (Transistor-Transistor-Lógico).

### 2.1.1 Componentes de la tarjeta de adquisición de datos

a) DAQCard-AI-16E-4 (figura 2.2), es la tarjeta de adquisición de datos que se conecta a la computadora en la ranura PCMCIA teniendo al otro extremo conectado el cable PSH68-68.



Figura 2.2 Tarjeta DAQCard

b) Cable PSHR68-68 (figura 2.3). Utilizado para conectar la DAQCard con el dispositivo SCB-68, cada conector del extremo tiene una forma trapezoidal para evitar una incorrecta conexión

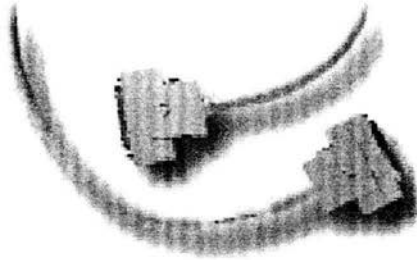


Figura 2.3 CablePSHR68-68

c) SCB-68 (figura 2.4). Dispositivos para conectar la señal obtenida de los sonares y la señal digital para activarlos. Funciona como una tableta de conexiones. Se conecta al cable PSHR68-68 para mandar y obtener las señales de la tarjeta DAQCard.

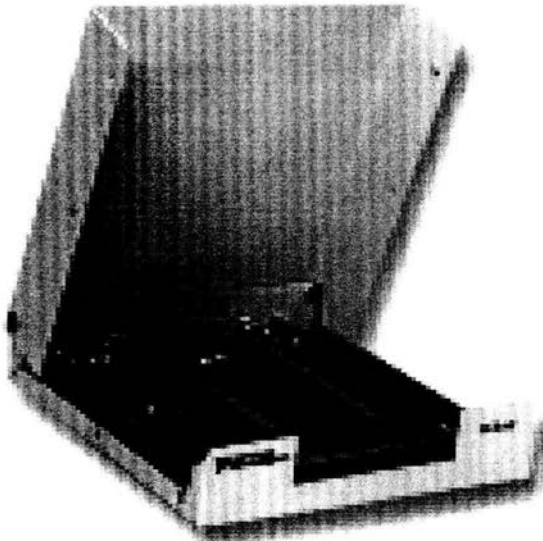


Figura 2.4 SCB-68

## 2.2 Identificación de las terminales del dispositivo SCB-68

El dispositivo SCB-68 tiene 68–pines para las conexiones digitales y analógicas. Los pines están agrupados para realizar una función específica y se les llama puertos. Para el objetivo de la tesis se utilizan un puerto analógico y otro digital. A continuación se describen la posición de los puertos.

### 2.2.1 Conexiones del dispositivo SCB-68

La figura 2.5 muestra la posición de pines para el dispositivo SCB-68, que forman los puertos utilizados en la tesis.

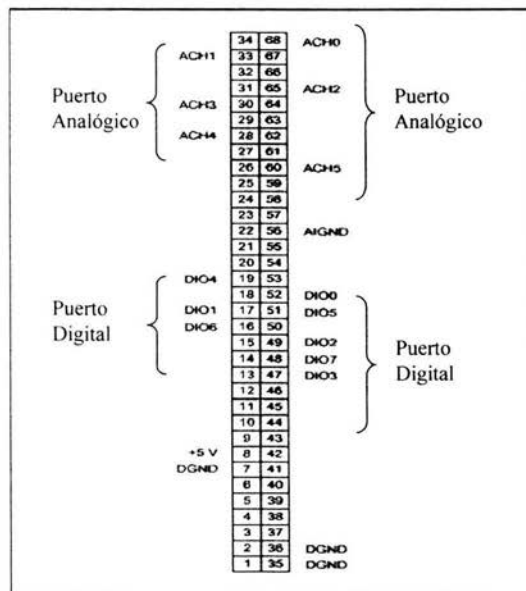


Figura 2.5 Pines del dispositivo SCB-68

La descripción de los pines utilizados para la captura de datos y activación de los sonares se muestran en la Tabla I:

Nombre	Referencia	Dirección	Descripción
AIGND	-	-	Entrada analógica de tierra- estos pines son puntos de referencia para los canales analógicos (ACH).
ACH<0..5>	AIGND	Entrada	Canales analógicos de entrada de 0 a 5 para las señales adquiridas de los sonares formando el puerto analógico de entrada.
DGND	-	-	Tierra digital-Este pin suministra la referencia para los canales digitales (DIO) y la fuente de poder de +5 VDC.
DIO<0..7>	DGND	Salida	Canales digitales de salida de 0 a 7 para la activación de los sonares formando el puerto digital de salida.
+5 V	DGND	Salida	Fuente de +5 VDC para la alimentación de los sonares.

Tabla I Descripción de pines. Referencia del manual DAQCard E Series User Manual

### 2.3 Instalación de los dispositivos

La tarjeta DAQCard tiene dos extremos uno tiene 68-pines para la ranura PCMCIA y 68-pines en el extremo para el cable PSHR68-68. Se inserta el extremo de PCMCIA en la ranura de la computadora hasta que el conector esté firme. En el otro extremo se inserta el cable PSHR68-68M. Este cable se unirá al cable SH6868 como se muestra en la figura 2.6.

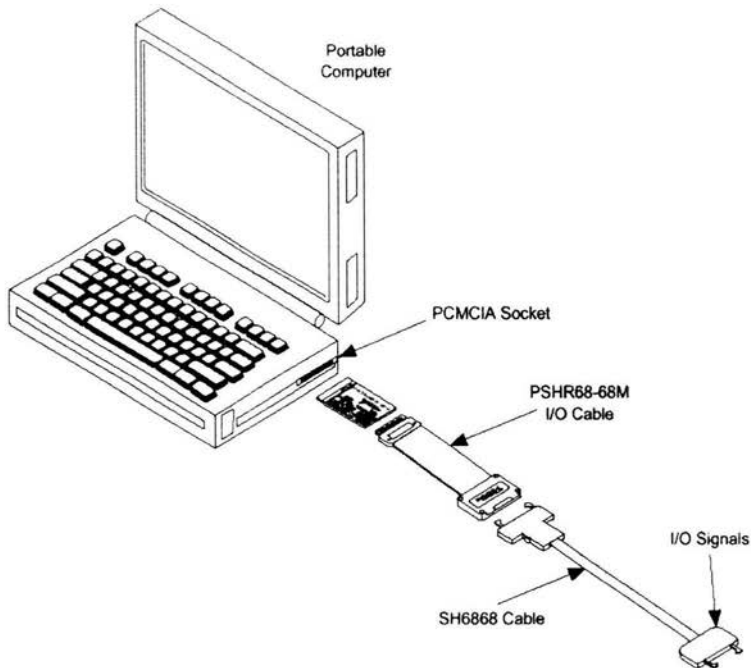


Figura 2.6 Conexión de la tarjeta DAQCard

Finalmente el conector del cable SH6868 se conecta al dispositivo SCB-68 para tener ensamblado los dispositivos para la adquisición de datos.

#### 2.4 Instalación de controladores de la tarjeta NI-DAQ

Los controladores de la tarjeta NI-DAQ se encuentran en los CD-ROM National Instruments NI-DAQ Data Acquisition Drivers Software que trae la tarjeta.

Los pasos son los siguientes:

1. Se inserta el CD-ROM 1.
2. Aparecerá una pantalla para instalar la tarjeta NI-DAQ (ver figura 2.7).



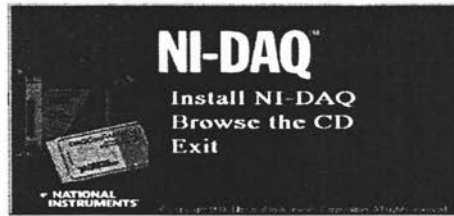


Figura 2.7 Ventana de instalación para la tarjeta DAQCard

3. Las figuras 2.8.a, b y c muestran las pantallas para la instalación correcta.



Figura 2.8.a Bienvenida de la instalación

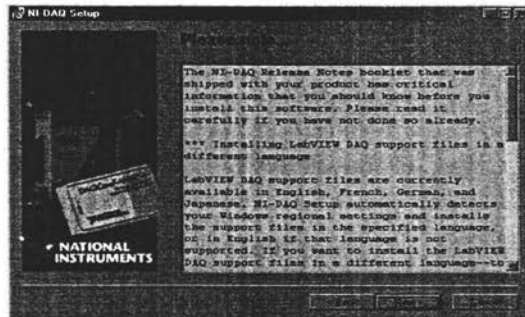


Figura 2.8.b Políticas de Nacional Instruments

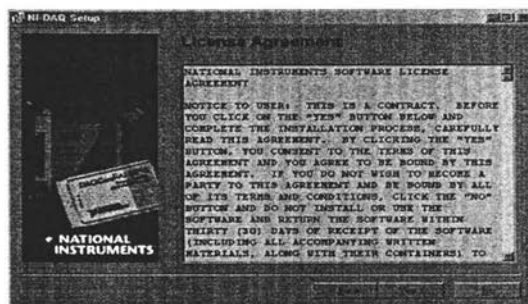


Figura 2.8.c Conformidad de Licencia

- En la pantalla mostrada en la figura 2.9 aparece la opción de instalación, por default se tiene seleccionado los controladores de la tarjeta, se debe agregar la opción que instale los archivos para la creación de aplicaciones utilizando Microsoft Visual C++.

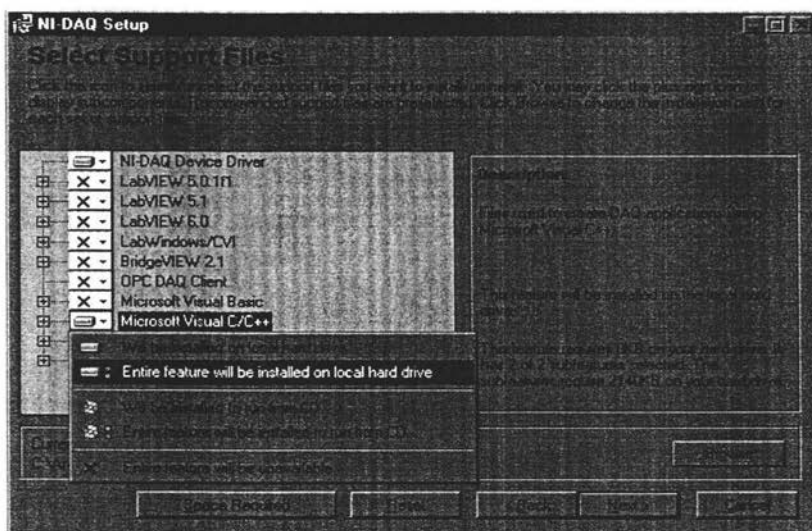


Figura 2.9 Selección de la aplicación Microsoft Visual C++

- Es requerido que después de terminada la instalación se reinicie la laptop para finalizar la instalación de la tarjeta NI-DAQ.

## 2.5 Configurar Microsoft Visual C++.

Para desarrollar una aplicación NI-DAQ se siguieron los siguientes pasos:

1. Microsoft Visual C++.
2. Abrir un nuevo proyecto en Visual C++ para manejar el código de la aplicación **File/New/File/Text File**. Se debe asignar un nombre en **File name** con la extensión **.c**.
3. Crear el código del programa. Incluyendo la cabecera del NI-DAQ que es el archivo `nidaqex.h`  
`#include "nidaqex.h"`
4. Compilar para crear el Workspace. **Build/Compile** Archivo.
5. Especificar el directorio que contiene los archivos de cabecera de NI-DAQ sobre **Project/Setting/C/C++**, escoger en Category **Preprocessor** en **Additional include directory**, escribir la ruta del directorio `\Include` sobre el directorio NI-DAQ en Archivos de Programa de National Instruments.
6. Importar las librerías NI-DAQ al proyecto **Project/Setting/Link**, escoger en Category **Input** en **Additional library path**, y escribir la ruta del directorio `\lib` sobre el directorio NI-DAQ en Archivos de Programa.
7. Construir la aplicación.

## 2.6 Sistema Digital

La DAQCard contiene ocho líneas digitales Entrada/Salida para propósitos de uso general. Cada línea se puede configurar individualmente por medio de software tanto de entrada como de salida.

### 2.6.1 Conexiones de las entradas digitales

Las entradas digitales son DIO<0..7> y DGND. Las señales DIO<0..7> forman el puerto DIO y DGND esta última es la tierra referenciada para el puerto. Se puede programar todas las líneas individualmente para ser entradas o salidas.

La figura 2.10 muestra la conexión para tres señales digitales típicas, donde el DIO <0 ..3> es configurado para entradas digital y el DIO <4 ..7> para señales de salida, para los fines buscados este puerto activara los sonares.

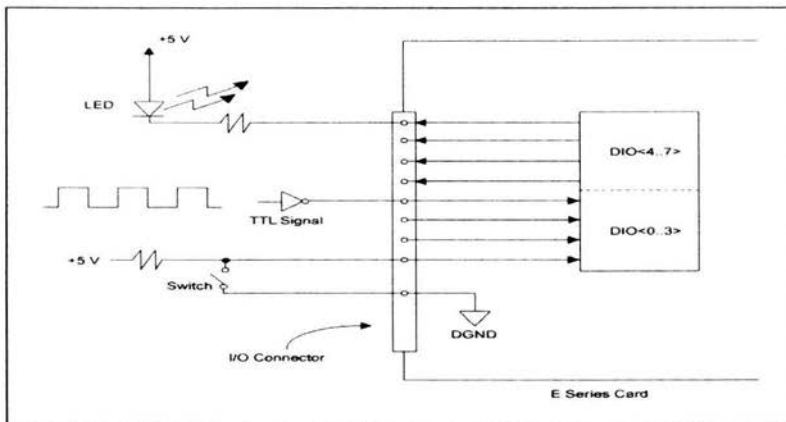


Figura 2.10 Puerto Digital

### 2.6.2 Configuración

Para la configuración del sistema digital es necesario conocer el puerto digital Entrada/Salida, para determinar el modo en que trabajara utilizando las funciones proporcionadas por la tarjeta.

### 2.6.3 Funciones digitales de salida.

Este dispositivo contiene ocho pines para el puerto digital de Entrada/Salida. Nosotros podemos controlar el conjunto de líneas digitales como un grupo para lectura o escritura según convenga. Configuramos el puerto de salida para fines de activación de los sonares. En NI-DAQ se refiere al puerto por un número. Muchos dispositivos etiquetan el puerto por una letra, el cero para el puerto A, el numero uno para el B y así sucesivamente. Sólo utilizaremos seis líneas de salida para los seis sonares, identificamos el puerto como cero, teniendo el siguiente mapeo con el número de línea. Tabla II

Numero de Bit	Numero de línea de salida
5	5
4	4
3	3
2	2
1	1
0	0 LSB Menos significativo.

Tabla II Relación del puerto digital con el arreglo de bits. Tabla generada durante la tesis.

#### **2.6.4 Funciones digitales de Entrada/Salida.**

Las funciones mas utilizadas para el puerto de salida digital son las siguientes:

DIG\_Prt\_Config – Configura el puerto de salida o de entrada.

DIG\_Out\_Prt – Especifica el valor de salida en el puerto especificado.

#### **2.6.5 Programa de prueba**

La primera prueba para conocer el funcionamiento de la salida digital fue obtener un pulso en un canal, utilizando las funciones contenidas en las librerías de la DAQ-Card.

La generación de un pulso en el puerto digital de salida es producido por el siguiente código en Microsoft Visual C++ (ver Apéndice A como Programa 1).

En el pin 52 DIO<0> se conectó un led utilizando la salida de 5 volts, en el pin 8, para realizar las pruebas del puerto digital de salida como se muestra en la figura 2.11.

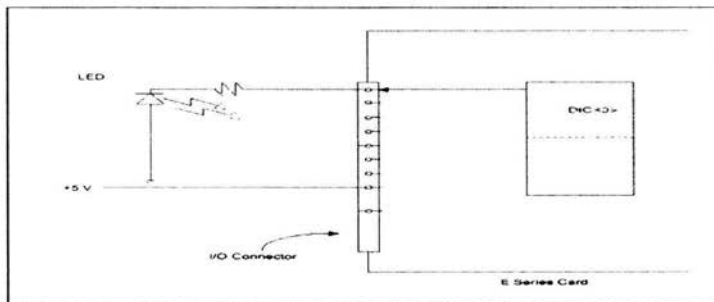


Figura 2.11 Led conectado al dispositivo SCB-68

## 2.7 Sistema Analógico

Es sistema analógico se utiliza para capturar señales no digitales y procesarlas por la tarjeta DAQCard.

### 2.7.1 Conexiones

Existen diferentes formas de referenciar las señales analógicas en la tarjeta para su captura y el procesamiento de las mismas.

### 2.7.2 Consideración de las conexiones

En el modo referenciado (RSE) y no-referenciado (NRSE), señal de entrada analógica de la DAQCard es referenciada a una tierra, que puede ser compartida con otras señales de entrada.

Se usa modo referenciado cuando las señales de entrada cumplen con las siguientes condiciones:

- La señal de entrada en nivel bajo ( $< 1 \text{ V}$ ).
- El cable de conexión de la señal de entrada al DAQCard están a menos de 3 m.
- La señal de entrada puede compartir un punto de la referencia común con otra señal.

Se puede configurar por medio de software, los canales de la DAQCard para dos tipos diferentes de conexión RSE y configuración de NRSE. Tabla III

<b>Configuración</b>	<b>Descripción</b>
RSE	Un canal configurado en modo RSE usa una línea de entrada la cual se conecta a la entrada positiva. La entrada negativa es internamente conectada a tierra utilizando el pin AIGND.
NRSE	Un canal configurado en modo NRSE usa una línea de entrada la cual se conecta a la entrada positiva. La entrada negativa se conecta al pine AISENSE.

Tabla III. Descripción de RSE y NRSE. Referencia del manual DAQCard E Series User Manual

En la conexión de los canales analógicos se utilizó la configuración RSE para tener las señales de entradas referenciadas por medio de AIGND.

Las señales de entrada analógicas son ACH <0 ..15> y AIGND. Las ACH <0 ..15> están ligadas a los 16 canales de entrada analógicos de la DAQCard. En modo referenciado y no-

referenciado, las señales conectadas a ACH <0 ..15> son dirigidas a la entrada positiva del DAQCard.

### 2.8 Polaridad y Rango de Entrada

Esta tarjeta tiene dos polaridades de entrada, unipolar y bipolar. La tarjeta tiene un rango de entrada unipolar de 10 V(0-10 V) y un rango bipolar de 10 V ( $\pm 5$  V), .

La ganancia que se puede programar por software en la tarjeta, incrementa su flexibilidad, a cada rango de señal entrada que pueda soportar el ADC le corresponde una ganancia determinada. La tarjeta tiene una ganancia de 0.5, 1, 2, 5, 10, 20, 50 y 100. La tabla IV muestra un conjunto de rangos de entrada y precisión acorde al rango configurado y la ganancia usada.

Configuración de rango	Ganancia	Rango de entrada	Precisión
0 a +10 V	1.0	0 a +10 V	2.44 mV
	2.0	0 a +5 V	1.22 mV
	5.0	0 a +2 V	488.28 $\mu$ V
	10.0	0 a +1 V	244.14 $\mu$ V
	20.0	0 a +500 mV	122.07 $\mu$ V
	50.0	0 a +200 mV	48.83 $\mu$ V
	100.0	0 a +100 mV	24.41 $\mu$ V
-5 a +5 V	0.5	-10 a +10 V	4.88 mV
	1.0	-5 a +5 V	2.44 mV
	2.0	-2.5 a +2.5 V	1.22 mV
	5.0	-1 a +1 V	488.28 $\mu$ V
	10.0	-500 a +500 mV	244.14 $\mu$ V
	20.0	-250 a +250 mV	122.07 $\mu$ V
	50.0	-100 a +100 mV	48.83 $\mu$ V
100.0	-50 a +50 mV	24.41 $\mu$ V	

Tabla IV Rangos actuales y medidas de precisión. Referencia del manual DAQCard E Series User Manual

### 2.9 Las Conexiones para el Señal Flotante (la Configuración de RSE)



La figura 2.12 muestra cómo conectar una señal fuente flotante a un canal dentro de una DAQCard con configuración en el modo de RSE.

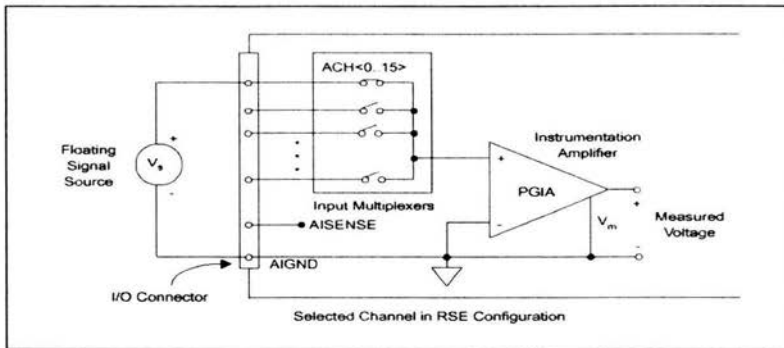


Figura 2.12 Modo referenciado

### 2.9.1 Conexiones de Alimentación

El dispositivo SCB-68 proporciona una fuente de alimentación con un rango de +4.65 a +5.25 VDC a 250 MA en el pin 8 con la tierra DGND en el pin 9 que serán usados para alimentar a los sonares.

### 2.10 Configuración

Antes de comenzar el desarrollo de la aplicación, se probó el dispositivo DAQ con el software Measurement & Automation Explorer.

Measurement & Automation Explorer es una aplicación para Windows que se usa para configurar los dispositivos DAQ de National Instruments sobre Windows 2000/NT/9x.

Esta aplicación se instala con el CD-ROM que incluye la DAQ-Card. Se crea un icono de acceso directo en el escritorio.

Las primera prueba con la DAQCard fue conectar un generador de señales en los pines ACH<0> y AIGND del SCB-68 (figura 2.13), para verificar con el software **Measurement Studio Software**, la adquisición de las señales.

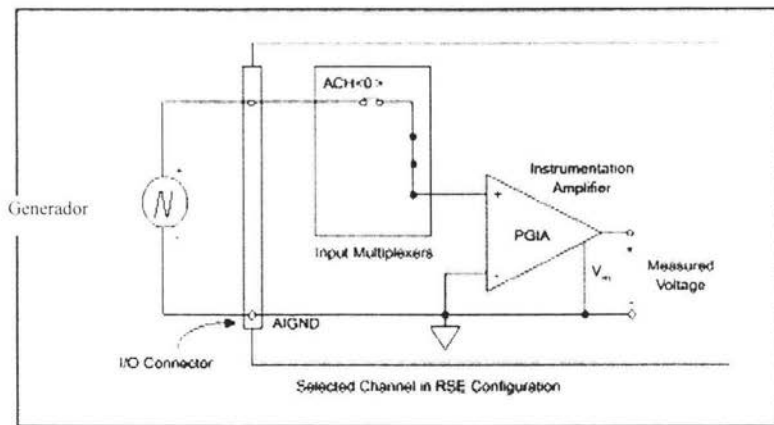


Figura 2.13 Generador de señales conectado al SCB-60

Mostrando las señales en la interfaz gráfica del **Measurement Studio Software** se comprobó el funcionamiento de la tarjeta de adquisición de datos (ver figura 2.14).

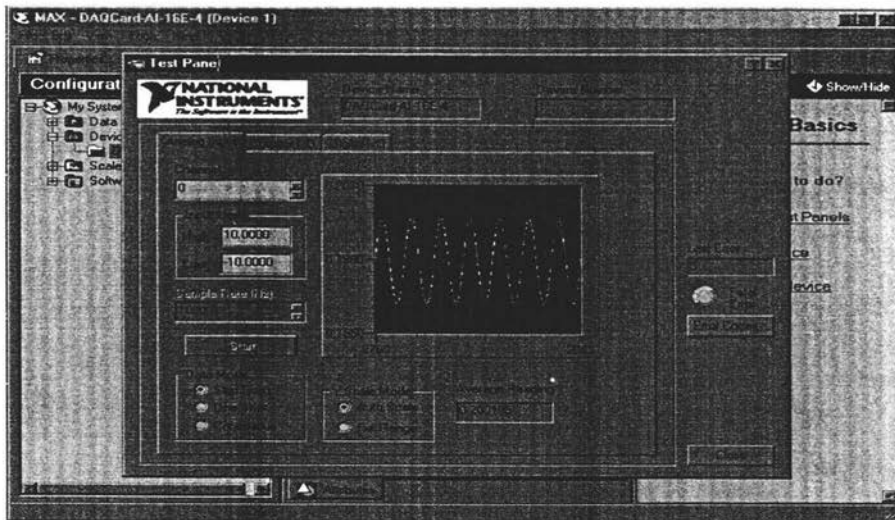


Figura 2.14 Funcionamiento del Measurement Studio Software

Las conexiones analógicas de entrada, configuradas por software, funciono correctamente.

Cabe mencionar que se pueden seleccionar diferentes configuraciones de entradas analógicas a través de la aplicación diseñada por el software para controlar el DAQCards.

### **2.10.1 Funciones analógicas de Entrada y Salida**

La tarjeta DAQCard incluye funciones para programarla, las utilizadas en la tesis son las siguientes:

`iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt)`: Convierte una velocidad de adquisición de datos en intervalos de muestreo.

`iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt)`; se establece la ganancia en uno, asignándose a la variable `iGain`, para tener un rango de entrada de 0 a +10V.

`iStatus = DAQ_Clear(iDevice)`: Cancela la última operación de adquisición de datos y reinicia los circuitos de adquisición.

### **2.12 Programa de prueba**

Utilizando las funciones contenidas en la librería del NI-DAQ se desarrolló un programa de prueba para la adquisición de datos por un puerto de entrada analógico, guardando los datos en el archivo "datos.dat". El código del programa se encuentra en el Apéndice A Programa 2.

El programa rastrea el puerto cero de entrada análogo, guardando 3,000 muestras en un archivo con el nombre datos.dat.

La visualización de los datos obtenidos se realiza con la plataforma MATLAB utilizando el siguiente script:

```
load datos.dat  
plot( datos(1:3500) )
```

Obteniendo imagen desplegada en la figura 2.15.

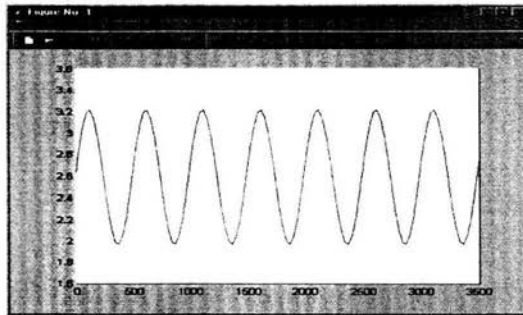


Figura 2.15 Visualización del archivo datos.dat

## CAPÍTULO III SONARES ULTRASÓNICOS

### 3.1 Ultrasonido.

Se entiende por ultrasonido a toda onda mecánica ó de presión cuya frecuencia es superior al rango auditivo del ser humano. Las frecuencias audibles para la mayoría de las personas, corresponden a frecuencias ubicadas entre los 20 Hz y los 16 kHz. (Llegando en algunos casos hasta los 20 kHz.). La energía se propaga, por tanto a través de las partículas individuales de la materia. El número de oscilaciones o vibraciones de las partículas nos dice si se trata de sonido perceptible por el oído humano o de ultrasonido.

Si bien las aplicaciones ultrasónicas abarcan campos tan diversos como la Ingeniería (Soldadura de plásticos, lavado por vibración, secado, sonar, etc.), la medicina (Ecógrafos, destrucción de cálculos, etc.). En esta tesis se hablarán del caso particular del ultrasonido aplicado a los sonares para la detección y localización de objetos.

En la inspección de materiales por ultrasonido, las frecuencias son, por regla general, notablemente más elevadas y oscilan entre 0.25 y 25 MHz. En principio, las ondas ultrasónicas pueden propagarse dentro de todos los medios donde existen fracciones de materia, o sea, átomos o moléculas capaces de vibrar, por lo que se propaga a través de gases, líquidos y sólidos. Las ondas ultrasónicas se propagan en cualquier material elástico cuando las partículas atómicas o moleculares de un material elástico son movidas desde su posición en equilibrio por cualquier fuerza aplicada; esfuerzos internos actúan para restablecer las partículas a su posición original, el desplazamiento de la materia que ocurre en las ondas ultrasónicas es extremadamente pequeño, la amplitud, el modo de vibración y la velocidad de las ondas difiere en los sólidos, líquidos y gases básicamente por la separación de las partículas en cada estado de la materia. Estas diferencias influyen las fuerzas de atracción entre las partículas y los medios elásticos de la materia.

### 3.2 Propiedades de las ondas ultrasónicas.

Las ondas ultrasónicas viajan a la misma velocidad que las ondas de sonido, lo único que cambia es la frecuencia. Los órdenes de magnitud para el aire son aproximadamente 343 m/s, para los sólidos de varios miles de metros por segundo (en el Aluminio está comprendido alrededor de 6,000 m/s) y de 1500 m/s en el Agua.

La atenuación de la onda ultrasónica es inversamente proporcional al cuadrado de la frecuencia.

Tanto las ondas electromagnéticas como las ondas acústicas poseen tres atributos que son: frecuencia, longitud de onda y velocidad de propagación.

$$v = \lambda * f$$

$v$  = velocidad de la onda en  $m/s$

$f$  = frecuencia en Hertz (Hz)

$\lambda$  = longitud de onda en metros (m).

También se puede expresar como:

$v = \lambda / T$  donde  $f = 1/T$

$T$  = período de la onda.

Otras propiedades relevantes de las ondas ultrasónicas son

- **la amplitud**
- **la potencia**
- **el medio de propagación.**

Existen dos formas típicas de propagación de las ondas ultrasónicas: estas pueden ser transversales o longitudinales. En la forma de propagación longitudinal, la vibración se

---

produce en la dirección de propagación de la onda, como en las ondas sonoras. En la forma de propagación transversal, las ondas se propagan en dirección ortogonal (en ángulos rectos) a la dirección de las zonas de compresión y de depresión. La propagación transversal ocurre cuando las ondas se propagan a lo largo de la superficie del medio, por ejemplo la superficie que se encuentra entre dos sólidos.

El modo de conversión está asociado con una pérdida significativa del nivel de la señal. En medios líquidos y tejidos se propagan bien las ondas longitudinales y, en cambio, las transversales son rápidamente amortiguadas.

### 3.3 Fenómenos de las señales ultrasónicas

Reflexión, refracción, difracción y dispersión son fenómenos que afectan todas las ondas y definen su comportamiento. Estos fenómenos suceden cuando las ondas inciden sobre medios con diferentes densidades, ver figura 3.1.

#### 3.3.1 Reflexión y refracción de ondas

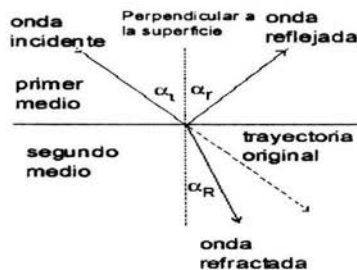


Figura 3.1 Incidencia de onda en diferentes densidades.

- Onda incidente.- la generada por el sonar.
- Onda reflejada.- se genera cuando la onda incidente encuentre un medio con distintas características acústicas.

- Onda refractada.- es la que absorbe el segundo medio

### 3.4 Transductores.

Un transductor es un dispositivo que convierte una señal de un tipo de energía en otra. La base es sencilla, se puede obtener la misma información de cualquier secuencia similar de oscilaciones, ya sean ondas sonoras (aire vibrando), vibraciones mecánicas de un sólido, corrientes y voltajes alternos en circuitos eléctricos, vibraciones de ondas electromagnéticas radiadas en el espacio en forma de ondas de radio, o las marcas permanentes grabadas en un disco o una cinta magnética.

#### 3.4.1 Características Transductores.

Para una transformación de energía de A a B, la conversión inversa, de B a A, se consigue con el mismo transductor operando en sentido opuesto. Los micrófonos y los altavoces constituyen una única familia de transductores operados en forma inversa. Para cada tipo particular de micrófono existe un altavoz que funciona, en sentido inverso, con los mismos principios físicos; en otras palabras, un micrófono sería un altavoz operado en sentido inverso. Por esta razón, ambos son transductores electroacústicos.

Existen diferentes tipos de transductores electroacústicos que se basan en leyes y propiedades físicas diferentes. A continuación vamos a describir los más importantes.

- **Electrostático o de condensador:** cuando la separación entre las placas del condensador varía, también varía su capacitancia, lo que a su vez provoca una diferencia de voltaje entre las placas. A la inversa, cuando una de las placas recibe una cantidad variable de carga eléctrica, la fuerza con la que atrae a la otra placa cambia, y como consecuencia ésta vibra. (*ver figura 3.2a*)
- **Piezoeléctrico:** cuando un material piezoeléctrico se contrae o expande dependiendo si es transmisor o receptor, aparece una cierta cantidad de carga positiva en uno de sus lados, y la misma cantidad de carga negativa en el lado



opuesto. Esta polarización puede ser recogida por un circuito eléctrico. A la inversa, cuando el material piezoeléctrico es polarizado por un circuito externo, se deforma. (ver figura 3.2b)

- **Magnético:** Existe una pieza de material magnético que se mueve mientras que la espira o arrollamiento permanece en reposo. (ver figura 3.2c)

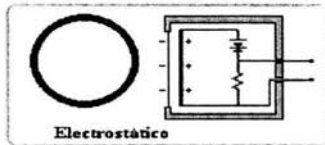


Fig 3.2a Transductor Electrostático

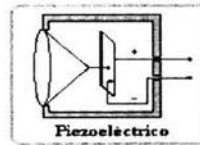


Fig. 3.2b Transductor Piezoeléctrico

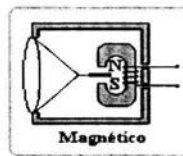


Fig. 3.2c Transductor Magnético

### 3.5 Sonares Ultrasónicos.

La mayoría de los sonares de ultrasonido se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia radial a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora, mediante la fórmula:

$$d = 1/2 v t$$

#### 3.5.1 Medición de distancia

La manera de obtener la distancia, es mediante la medición del tiempo de vuelo. Si se toma en cuenta T0 como el tiempo en el cual una ráfaga de pulsos es transmitida y T1 el tiempo

en el cual la ráfaga o parte de ella es recibida, entonces la distancia  $d$  de la figura mostrada entre el sonar y el objeto se determina de la siguiente forma, ver figura 3.3:

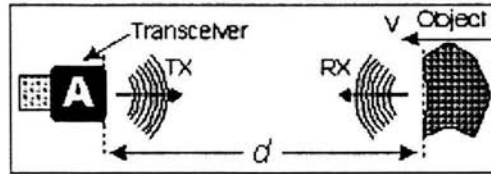


Figura 3.3 distancia "d" entre un transductor y un objeto (medición del tiempo de vuelo de una ráfaga de pulsos TX).

$$d = 0.5 * v * ( T1 - T0 )$$

### 3.6 Unidad de Medición Polaroid 6500

La serie 6500 de Polaroid no necesita una interfaz adicional para poder controlar el transductor Polaroid. Este modulo, con una simple interfaz es capaz de detectar un objeto a una distancia mínima de 30cm y como a una distancia máxima de 10mts (el máximo alcance depende del tamaño y forma del objeto que refleja el eco).

La unidad de medición de los transductores ultrasónicos Polaroid, tiene dos componentes principales: un transductor acústico y la tarjeta de circuitos de medición. Ver figuras 3.4a y 3.4b.

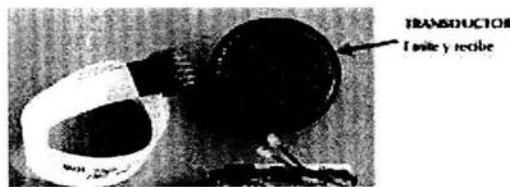


Figura 3.4a Transductor POLAROID 6500

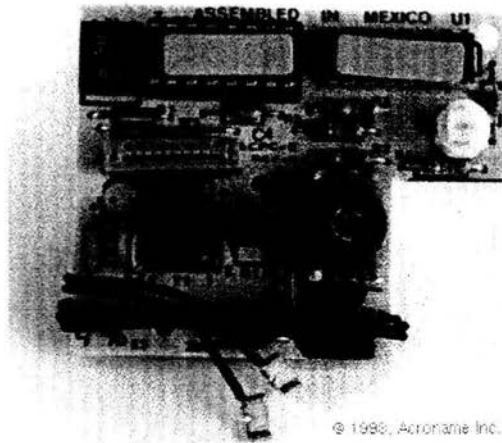


Figura 3.4b Tarjeta de Medición POLAROID 6500

Cuando la unidad está en operación, un pulso es transmitido en el medio de propagación y el eco puede ser detectado. El lapso de tiempo entre la transmisión inicial y la detección del eco, puede ser convertida en distancia teniendo en consideración la velocidad del sonido en el medio de propagación.

El principal componente en el sistema de medición de distancia Polaroid, es el transductor, el cual actúa como transmisor y receptor (*ver figuras 3.4 a y 3.4 b*). Está diseñado para transmitir la señal saliente y también para funcionar como un micrófono electrostático para así recibir la señal reflejada (eco). El diámetro del transductor determina el ángulo de tolerancia, durante la operación de transmitir y recibir. Una hoja especial de papel Aluminio de Polaroid es adherida sobre la lámina con ranuras, formando así el elemento móvil, el cual transforma la energía eléctrica en ondas de sonido y el eco reflejado lo regresa a energía eléctrica.

La parte metálica posterior con ranuras en contacto con el lado aislado de la hoja de aluminio forma un capacitor el cual, cuando es cargado, proporciona una fuerza electrostática a la hoja de aluminio. La hoja de aluminio es flexible, capaz de resistir a ambientes ásperos y es un excelente conductor eléctrico.

Cuando la unidad es activada, el transductor emite un pulso de ultrasonido, luego espera a recibir el eco reflejado.

La tarjeta con los circuitos de medición (*ver figura 3.4b*) controla el modo de operación (receptor/transmisor) del transductor. Comprende tres secciones principales, las cuales controlan la operación del transductor y permiten que la información pueda ser usada como se desea. Entre dichas secciones están: un circuito digital, un circuito analógico y una sección de alimentación.

Alimentado por una fuente de 5 VDC, capaz de manejar una corriente de 2.5 A/ms. La señal para activar la tarjeta POLAROID que genera el pulso ultrasónico debe ser proporcionada por un circuito externo.

Un reloj cerámico controlado por resonancia en el circuito digital genera el pulso transmitido por el transductor.

Después de haberse generado el pulso externo, el modo de operación del transductor cambia de transmisor a receptor, para detectar el eco de regreso.

Una vez recibido el eco, el transductor convierte la energía mecánica a energía eléctrica, la cual es amplificada por el circuito analógico, luego detectada por el circuito digital para producir la señal recibida. En la figura 3.5 se muestra la unidad de medición Polaroid internamente.

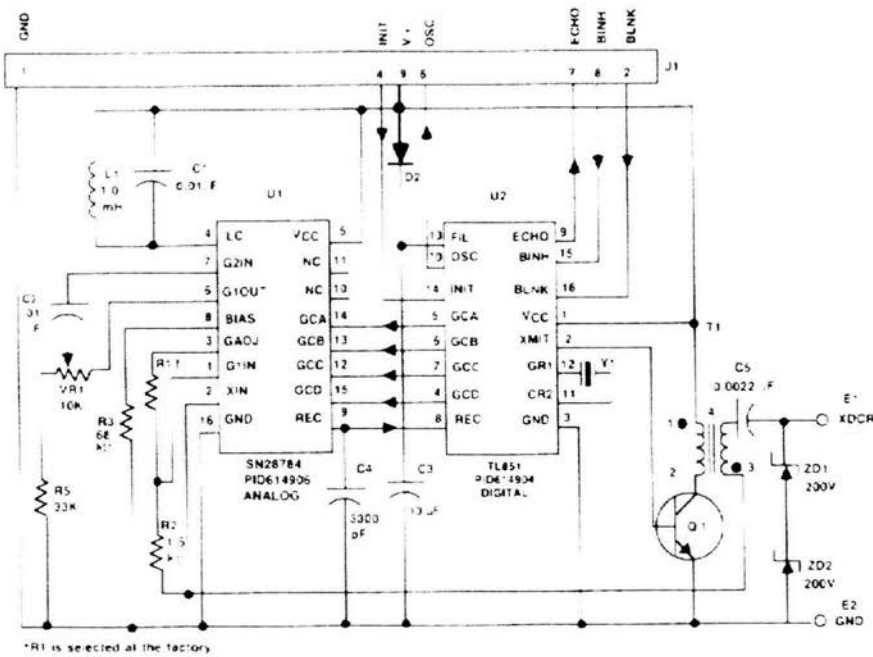


Figura 3.5 Diagrama Interno de la Tarjeta POLAROID 6500

### 3.6.1 Descripción de las Señales.

Existen nueve conectores para la señal digital de entrada y salida en el módulo Polaroid 6500. A continuación se listan los números de las señales que corresponden a los conectores de la tarjeta Polaroid.

- 1) GND
- 2) BLNK
- 3) Sin conexión
- 4) INIT
- 5) Sin conexión
- 6) OSC
- 7) ECHO
- 8) BINH
- 9) Vcc

Para iniciar el pulso de activación en la tarjeta, un pulso positivo de 10 ms debe ser aplicado a la tarjeta Polaroid 6500 al conector 4) INIT.

El transductor del sonar es un elemento que actúa como transmisor, para emitir el “chirp” ultrasónico, y en un par de milisegundos el mismo elemento se configura así mismo como un receptor. Existen limitaciones, tanto físicas como de software, que determinan que tan rápido el transductor puede realizar este cambio. La transformación no puede ser instantánea, debido a la membrana electrostática que está dentro del transductor, que vibra para generar la señal, debe tomar un momento para estar limpia, y así escuchar el eco. Si se pone en modo de escuchar antes que el transductor haya vibrado, detectará las vibraciones de su propio pulso, esto provocará que se corrompan los ecos externos detectados subsecuentemente.

La tarjeta Polaroid resuelve el problema generando un período llamado “blanking” inmediatamente después de que el pulso “chirp” es iniciado. La configuración original para este período es alrededor de 2.4 ms. Después de este retraso, el sonar está listo para cualquier eco entrante. El problema es un período ciego de 2.5 ms que equivale aproximadamente a 60 cm. Esto indica que el sonar no detectará objetos a menos de 30 cm si se deja la configuración inicial del “blanking”.

La señal del conector número ocho BINH inhibe la opción de “blanking”. Cuando este conector se alimenta con un pulso positivo, el modo “blanking” del sonar automáticamente se apaga, y la tarjeta es forzada a ponerse en el modo “escucha” permitiendo detectar objetos a 30 cm de distancia como mínimo. Es posible activar el modo INIT y el modo BINH al mismo tiempo, si esto ocurre, ningún eco externo de un objeto detectado será recibido por la tarjeta simplemente reconoce los ecos de su misma señal. Este pin está disponible si se desea detectar objetos a una distancia menor a 30 cm, el problema es que si el sonar se configura para detectar objetos a una distancia más corta, se tendrá el problema que el sonar detecte su propia señal emitida. Una calibración de mayor cuidado se requiere para realizar esta función.

Una vez que la señal de activación del sonar es generada, la energía ultrasónica radiada del sonar, hace contacto con el objeto en el campo visible del sonar y la energía reflejada o eco regresa hacia la dirección del sonar. Si el transductor detecta la energía reflejada, la señal ECHO se pone en un nivel alto. Esto es trabajo de la unidad de medición para medir la diferencia en tiempo entre la señal de inicio y la señal de regreso, la diferencia de tiempo puede ser usada para calcular la distancia radial

Cuando se activa la señal de inicio INIT se pone en alto, permanece así alrededor de 100 ms, la señal de eco es detectada en cualquier tiempo dentro de una ventana de 100 ms. Cuando la señal INIT vuelve a un estado bajo, la señal de eco es puesta en un nivel bajo, y el sistema esta listo para el siguiente pulso.

El método de operación es suficiente para el objetivo del proyecto, la tarjeta POLAROID puede ser configurada con características adicionales. En el modo de operación, también llamado "Modo de simple Eco"[4], después de que el primer eco es detectado por el conector ECHO, se queda en alto hasta que el pulso de inicio INIT se termina. Manteniendo la señal ECHO en alto, permite bloquear la detección de ecos adicionales que pueden ser detectados dentro de la ventana de 100 ms de la señal INIT, los llamados ecos de rebote, que son innecesarios para detectar un objeto. El conector ECHO recibe una señal para ponerlo en un nivel bajo dentro del tiempo de los 100 ms, para continuar escuchando ecos de otros objetos detectados.

Una de las características de la tarjeta de medición POLAROID es que la señal se puede obtener en su forma cruda tomando la señal antes de que sea procesada por el circuito TL851 ver figura 3.5, de esta manera se obtiene mayor información del eco reflejado producida por el objeto

### **3.7 Arreglo de sonares.**

Normalmente se combinan varios sonares para tener un rango mayor de visibilidad. Los arreglos pueden ser de forma radial, lineal, curvos, movibles o estáticos.

Para el proyecto se utilizó un arreglo lineal estático de seis sonares, para tener un rango mayor en la visibilidad del objeto. El rango visible de aproximadamente  $30^\circ$ , por cada transductor se muestra en la figura. 3.6

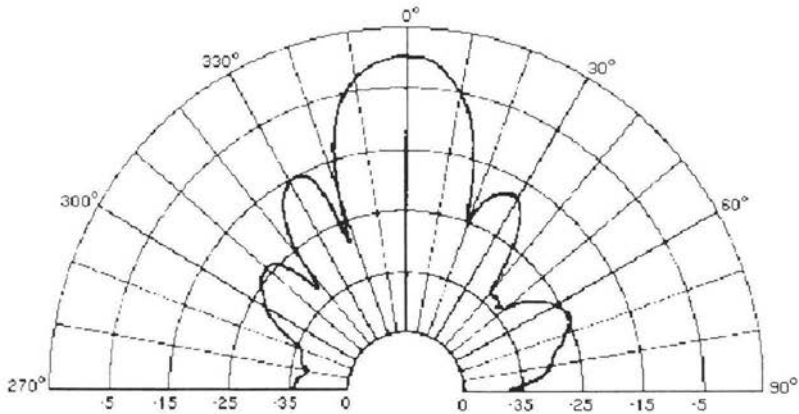


Figura 3.6 Rango Visible del Transductor POLAROID

El arreglo lineal de sonares se muestra en la figura 3.7

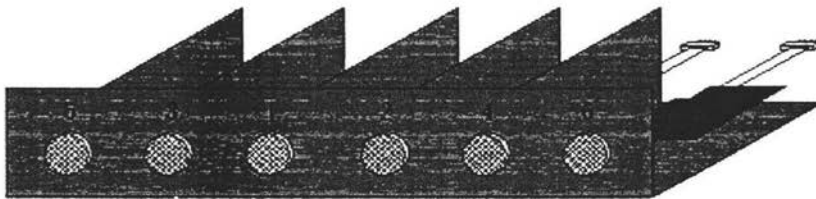


Figura3.7 Arreglo de Sonares

Se configuró el arreglo de sonares para que tuviera dos tipos de uso:

1.- *Configuración individual.*- Cuando solo sonar envía el pulso de ultrasonido y todos los sonares escuchan el eco generado por el objeto, donde el sonar elegido fue el sonar1.



2.- *Configuración serie.*- Cuando los seis sonares envían un pulso de ultrasonido en forma secuencial, empezando por el sonar1, sonar2, sonar3 hasta el sonar 6, y cada sonar recibe el eco reflejado.

La idea de realizar esos dos tipos de configuraciones era para tener los dos tipos de resultados y compararlos, dichas pruebas fueron realizadas y se explican en detalle en el Capítulo 4. La importancia de usar un arreglo de sonares es que se puede obtener mayor información del objeto que se desea detectar.

En el caso de tener un sonar, tendría que estar en movimiento sobre una trayectoria conocida si un robot móvil lo fuera a usar, tendría que tener una trayectoria radial para poder recopilar la mayor información posible del objeto. Para este proyecto se contaba con seis sonares por eso se optó por usar un arreglo lineal, las dos configuraciones realizadas al proyecto fueron necesarias para poder determinar de cual se obtenía mayor información y cual era mas estable en el capítulo cuatro se explica en detalle.

La región visible del arreglo de sonares es mayor radialmente (en cuanto a distancia longitudinal no cambia) teniendo como máximo 9 m y como mínimo 30 cm.

## **CAPÍTULO IV      DESARROLLO**

### **4.1 Introducción**

Este capítulo describe en forma detallada la implementación del sistema de sonares. El sistema de sonares consta de dos partes: a) Arreglo de sonares- los seis sonares polaroid fueron colocados en un arreglo lineal horizontal para tener mayor información del objeto detectado. b) Tarjeta de adquisición de datos- la tarjeta es utilizada para activar los sonares y capturar los ecos producidos por el objeto detectado.

### **4.2 Sonares Polaroid**

La mayoría de los sonares de ultrasonido se basan en la emisión de un pulso de ultrasonido cuyo campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia radial a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora.

#### **4.2.1 Pruebas de funcionamiento de sonares ultrasónicos**

La prueba que se realizó a cada sonar para verificar su funcionamiento es la siguiente:

1. La alimentación es un generador de funciones, con una señal de corriente directa de 5 V, conectando el pin 1 a Tierra y el pin 9 a 5 V.
2. Un osciloscopio para observar el eco obtenido por el sonar se conectó a los pines 1 GND y 6 OSC, como se explicó en el capítulo 3. Este dispositivo contiene dos canales: 1) Conectar la señal que se quiere observar, 2) Señal de disparo, cuando ésta se active el osciloscopio captura la señal en el primer canal en ese instante. Es necesario utilizar esta configuración debido a que no sería perceptible el eco detectado por el poco tiempo que dura esta señal.
3. Los pines 2 BLNK y 8 BINH se conectaron a tierra.

4. La señal INIT, para activar al sonar, se aplica manualmente con un alambre tomando los 5 V del generador de funciones.

Se probó el funcionamiento de los sonares con la señal BINH en alto para ver si era posible detectar objetos a una distancia menor de 30 cm. El eco capturado por el osciloscopio estaba distorsionado por esta razón se decidió tener la señal conectada a tierra.

#### 4.2.2 Arreglo de sonares

En el contexto de la tesis se plantea un sistema sonarial para detectar y localizar los objetos del entorno, con el fin de avisar al robot de la situación aproximada de éstos para evitar colisiones o para utilizarlos como referencias en su desplazamiento. Dados los movimientos de avance y giro del robot, se necesita disponer de un sistema de detección al frente del mismo. Se decidió que el arreglo para los sonares fuera lineal y en forma horizontal para poder obtener la posición bidimensional del objeto detectado (ver figura 4.1).

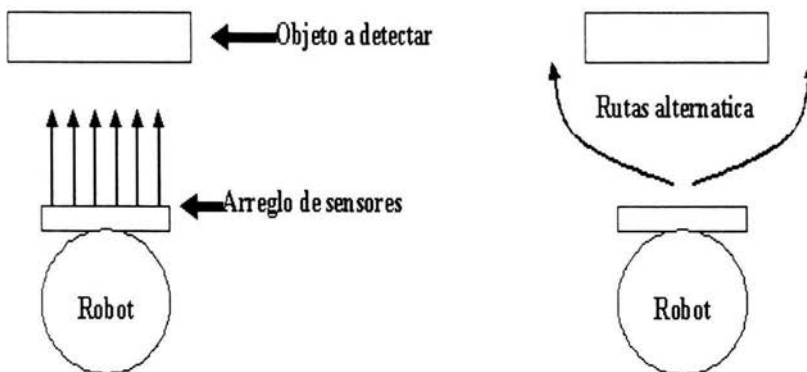


Figura 4.1 Posición del arreglo de sonares y el obstáculo

En este arreglo de sonares se ha optado por asignarles un número del uno al seis para su identificación, selectivamente se activarán proporcionando información de distancia de los objetos más próximos detectados en su sector, (ver Figura 4.2).

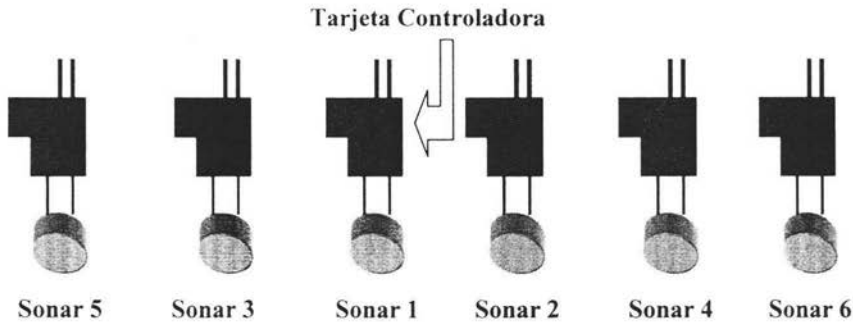


Figura 4.2 Orden de posición de los sonares

El soporte para el arreglo de sonares es de madera debido a su costo y por sus propiedades acústicas. La madera por su constitución se emplea para el aislamiento acústico, es decir, para impedir o dificultar el paso del sonido. La madera que más se emplea en estos casos es la de pino.

En específico se utilizó el triplay por su fuerza excepcional y alta resistencia, la figura 4.3 muestra el arreglo utilizado.

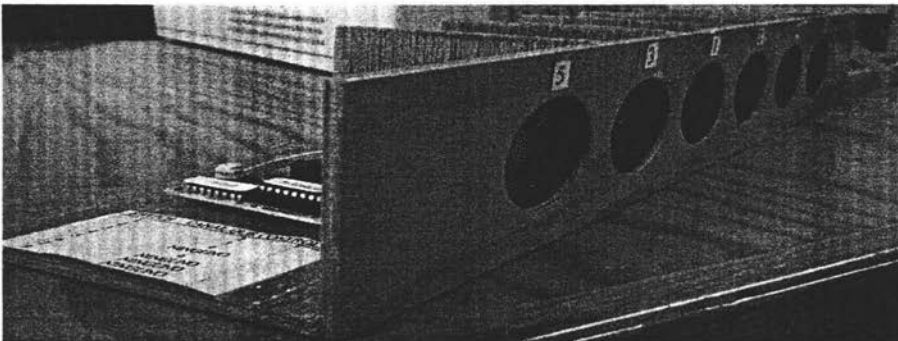


Figura 4.3 Arreglo de sonares

### 4.2.3 Modificaciones a la unidad de medición Polaroid #6500

La tarjeta del sonar recibe las señales de activación y ecos. Ésta incluye tres secciones importantes las cuales controlan la operación del transductor y permiten hacer uso de la información recolectada para los fines deseados. Entre estas tres secciones se encuentra el circuito digital, el circuito analógico y la sección de alimentación.

Los seis sonares obtienen la información de la energía reflejada, por la presencia de objetos ó bien por inhomogeneidades en el medio de transmisión. Cada sonar al recibir la señal INIT comienza a transmitir los 16 pulsos de ultrasonido a una frecuencia de 49.4 kHz, después que transcurren 2.38 ms el transductor pasa a modo receptor, este tiempo se conoce como INTERNAL BLANKING. El diagrama de tiempos de la señales se muestra en la figura 4.4.

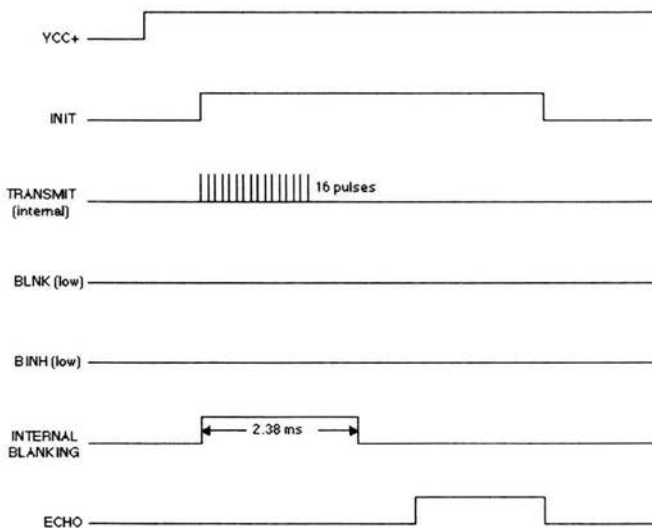


Figura 4.4 Diagrama de tiempo de las señales. Al activarse INIT se comienza la transmisión de los 16 pulsos, esperando 2.38 ms en espera de algún eco.

En caso que cada señal INIT de los sonares se active al mismo tiempo, los ecos obtenidos de un objeto detectado estarán alterados por el envío simultáneo de los pulsos de ultrasonido de los seis sonares.

Es requerido que todos los sonares se activen con la señal INIT siguiendo el diagrama de tiempos de las señales pero que sólo un sonar envíe los pulsos de ultrasonido como se muestra en la figura 4.5.

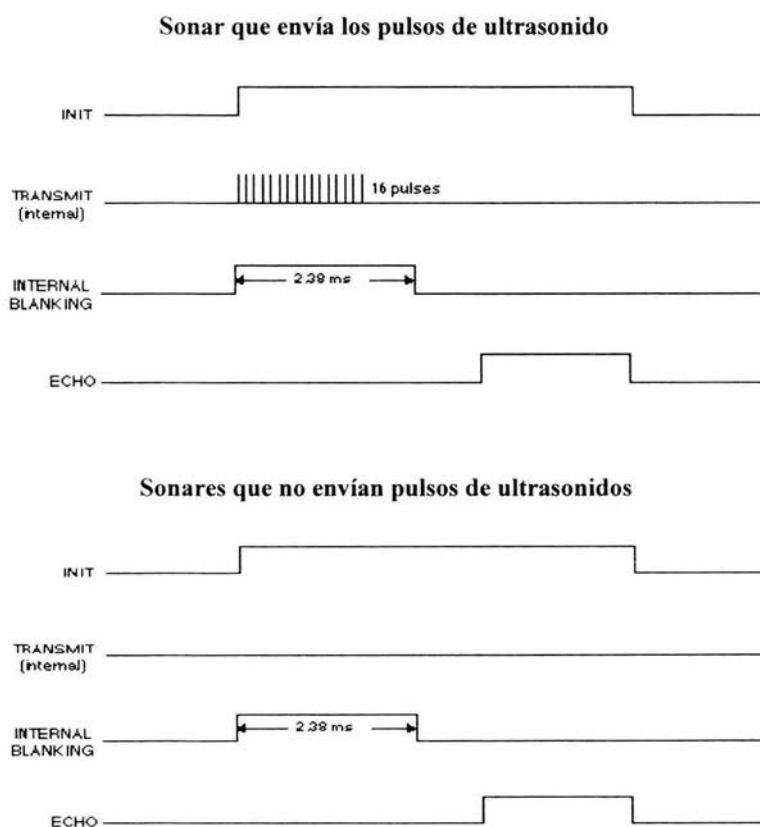


Figura 4.5 Diagrama de tiempos de las señales de los sonares, uno envía y todos reciben.

Para que un sonar Polaroid conserve el diagrama de tiempos, y que no transmita los pulsos ultrasónicos, se tuvo que realizar una modificación en la tarjeta controladora #6500.

El circuito U2 en el pin 2 (XMIT), se transmite los 16 pulsos de la señal TRANSMIT de la base al colector del transistor Q1 después de recibir la señal INIT en el pin 14. Abriendo el circuito en la base del transistor, se evita que los pulsos lleguen al transductor sin cambio alguno el diagrama de tiempo de las señales (ver Figura 4.6).

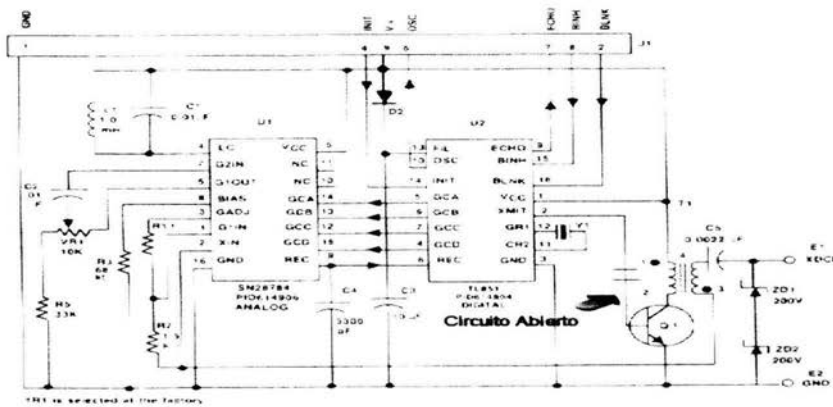


Figura 4.6 Modificaciones a la tarjeta #6500

La arquitectura elaborada para realizar la excitación y obtener el eco de cada uno de los sonares responde a dos tipos de configuración:

- a) *Individual* El sonar transmite y recibe (modo pulso-eco)-En principio se pensó tener únicamente la configuración de que cada sonar transmite y obtiene el eco del objeto detectado siguiendo el orden del uno al seis.
- b) *Serie* Un solo sonar transmite y los seis recibe- puede tenerse la configuración de que el sonar uno sea el único que transmite y se obtienen los ecos por los seis sonares en orden consecutivo.

Para la configuración *Serie* el sonar 1 no se modificó la tarjeta controladora. Se conectó un alambre a la base del transistor y otro al pin 2 del circuito U2 para conectarlos a un DIP Switch, para abrir y cerrar el circuito en el transistor de la tarjeta controladora, dependiendo la configuración que se desee (ver figura 4.7).

El DIP Switch se colocó en una tarjeta para unirlos con los alambres conectados a la tarjeta controladora de cada sonar, disminuyendo el número de alambres.

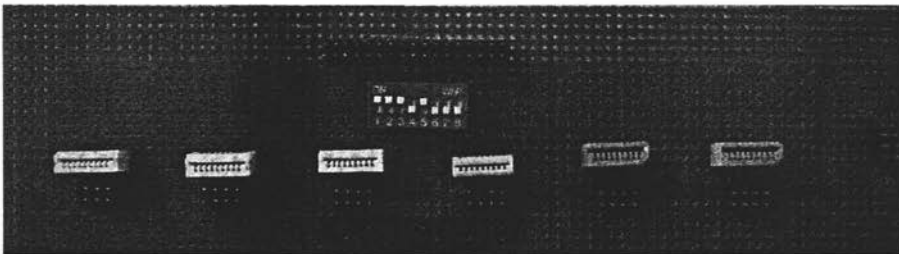


Figura 4.7 DIP Switch

La configuración *Serie* puede ser la mejor opción para obtener información de los ecos del objeto detectado y analizarlos con algoritmos para reconstruir las dimensiones. La información obtenida puede ser incrementada si se utiliza una tarjeta de mayor capacidad, ya que la frecuencia de muestreo se divide el número de sonares.

#### 4.3 Conexiones entre los cables de control de los sonares

El cable de cada sonar con las señales de control, ECHO y la de alimentación se conectaron a la tarjeta donde se encuentra el DIP Switch para poder tener todas las conexiones reunidas en un arreglo de pines en forma ordenada para su mejor manejo como se muestra en la Figura 4.8.



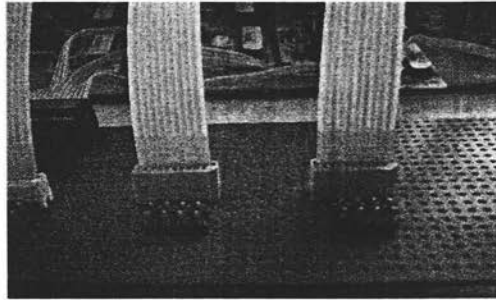


Figura 4.8 Conexión de cables

#### 4.4 Acoplamiento del arreglo de sonares y la tarjeta DaqCard

En la tableta donde se reúnen todas las conexiones del arreglo de sonares y el DipSwitch se conecta un cable al arreglo de pines hacia el dispositivo SCB-68. Las señales que se enviarán desde la tarjeta son: el INIT y las de alimentación para los sonares, haciendo uso del puerto digital de salida. Los pines utilizados en el SCB-68 fueron los siguientes:

Señal	Puerto Digital	Pin
INIT del sonar 1	D0	52
INIT del sonar 2	D1	17
INIT del sonar 3	D2	49
INIT del sonar 4	D3	47
INIT del sonar 5	D4	19
INIT del sonar 6	D5	51
5 V	+5V	8
Tierra	DGND	7

Tabla generada durante la tesis

La tarjeta DaqCard captura los eco detectados por los sonares, para este fin se utiliza el puerto analógico de entrada teniendo la siguiente asignación:

Señal	Puerto Analógico	Pin
ECHO de sonar 1	ACH0	68
ECHO de sonar 2	ACH1	33
ECHO de sonar 3	ACH2	65
ECHO de sonar 4	ACH3	30
ECHO de sonar 5	ACH4	28
ECHO de sonar 6	ACH5	60
Tierra del puerto análogo	AIGND	56

Tabla generada durante la tesis

La configuración final para el envío de señales y la captura de los ecos en el dispositivo SCB-68 se muestra en la figura 4.9

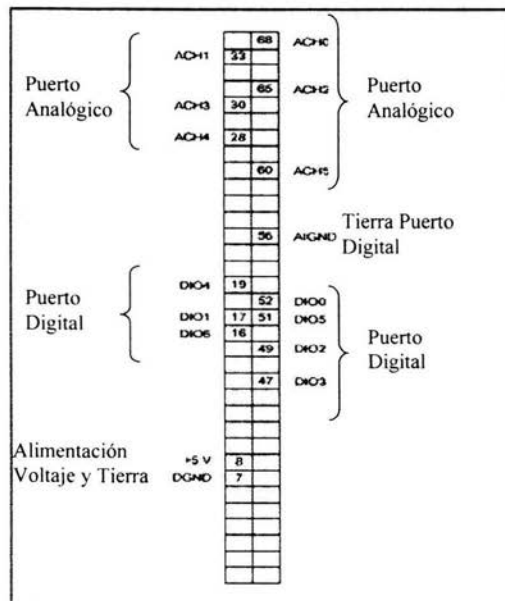


Figura 4.9 Configuración de puertos

#### 4.5 Activación de sonares

La activación de los seis sonares se realizará enviando una secuencia de pulsos en el puerto digital de salida de 7 bits de la tarjeta DaqCard, cada pulso corresponde a un sonar. Tabla 4.1

Puerto Digital	Sonar
Bits 0	1
Bits 0	2
Bits 0	3
Bits 0	4
Bits 0	5
Bits 0	6
Secuencia 000000	

Tabla 4.1 Correspondencia entre el puerto digital y los sonares. Tabla generada durante la tesis

#### 4.6 Configuración Individual

En este tipo de secuencia es necesario que no esté desconectada la base del transistor Q1. El DipSwitch tendrá que cerrar el circuito para tener la tarjeta #6500 sin cambio, como se muestra en la figura 4.10.

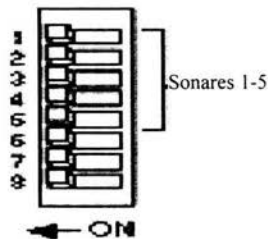


Figura 4.10 Configuración *Individual*

La secuencia de pulsos que se utilizó para activar los sonares en el modo en que cada sonar envía el ultrasonido y recibe el eco se muestra en la tabla 4.4

	Sonar 6	Sonar 5	Sonar 4	Sonar 3	Sonar 2	Sonar 1
<b>Secuencia</b>	0	0	0	0	0	1
	0	0	0	0	1	0
	0	0	0	1	0	0
	0	0	1	0	0	0
	0	1	0	0	0	0
	1	0	0	0	0	0

Tabla 4.4 Secuencia Binaria para activar sonares en configuración *Individual*.

Tabla generada durante la tesis.

La secuencia de pulsos representan un número binario, para enviarlo por el puerto digital de salida se envía el número binario en su forma decimal, la secuencia de números se muestra en la Tabla4.5.

Numero Binario	Número Decimal
000001	1
000010	2
000100	4
001000	8
010000	16
100000	32

Tabla 4.5 Correspondencia entre los números binarios y decimales en la activación de sonares. Tabla generada durante la tesis.

El programa que envía los pulsos se encuentra en al Apéndice B Programa 1. El programa despliega en la pantalla la secuencia de números que envía por el puerto de salida para corroborar la secuencia del envío de pulsos.

#### 4.7 Configuración Serie

En este tipo de secuencia es necesario que la base del transistor esté abierta para evitar el envío de los 16 pulsos de ultrasonido de los sonares 2 al 6, el DipSwitch tendrá que abrir el circuito colocándolo como se muestra en la figura 4.11.

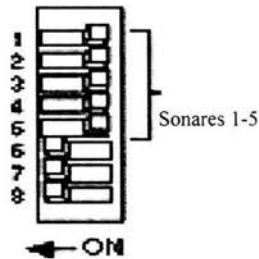


Figura 4.11 Configuración Serie

La siguiente secuencia muestra la asignación para cada sonar cuando únicamente el sonar 1 envía el pulso ultrasónico y en secuencia ascendente se va capturando los ecos detectados (Tabla 4.2).

	Sonar 6	Sonar 5	Sonar 4	Sonar 3	Sonar 2	Sonar 1
Secuencia	0	0	0	0	0	1
	0	0	0	0	1	1
	0	0	0	1	0	1
	0	0	1	0	0	1
	0	1	0	0	0	1
	1	0	0	0	0	1

Tabla 4.2 Secuencia de activación de sonares con configuración *serie*. Tabla generada durante la tesis.

De igual forma que la configuración individual los pulsos representan un número binario, para enviarlo por el puerto digital de salida, se envía el número binario en su forma decimal, la secuencia de números se muestra en la Tabla 4.3.

<b>Numero Binario</b>	<b>Número Decimal</b>
000001	1
000011	3
000101	5
001001	9
010001	17
100001	33

Tabla 4.3 correspondencia entre números binarios y decimales en la activación de sonares.

Tabla generada durante la tesis.

El programa que envía los pulsos se encuentra en el Apéndice B Programa 2. El programa despliega en pantalla la secuencia de números que envía por el puerto de salida para corroborar la secuencia del envío de pulsos.

#### **4.8 Captura de ecos**

Los sonares Polaroid tardan 2.38 ms para cambiar de emisor a receptor, esta característica de los sonares obliga a tener un programa que captura los ecos detectados por los sonares tan pronto y se activen con los programas de secuencia individual ó secuencia serie. El diagrama de flujo de la figura 4.12 muestra el proceso de activación y captura de ecos de los sonares.

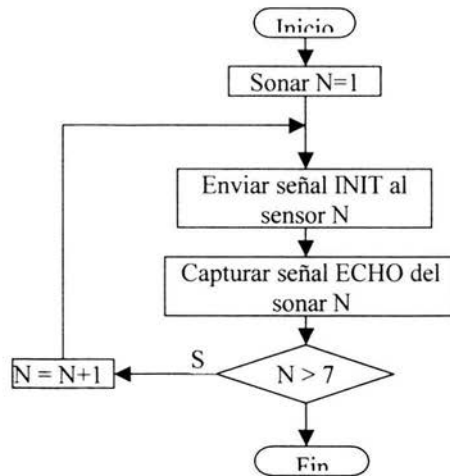


Figura 4.12 Diagrama de Flujo de activación de los sonares y captura de ecos.

El programa de captura de eco se encuentra en el Apéndice B como Programa 3. Este programa debe capturar el eco detectado del canal que se le indique por medio de la variable *iChan* y guardarlo en el archivo *datos.dat* para su análisis.

En la activación y captura de ecos se utilizaron los programas realizados en el capítulo 2, se unieron para obtener un solo programa, el Programa 3 guarda el eco del canal que se le indique y el programa que activa los sonares son Programa 1 ó Programa 2.

Se realizaron 2 formas de guardar los ecos:

- La primera consiste en guardar el eco detectado por cada sonar en un archivo diferente, identificado con el número del sonar.
- La segunda opción, los ecos son guardados en el mismo archivo consecutivamente, sabiendo que las muestras guardadas para cada eco, son de 3000. Por el objetivo de la tesis se decidió por la segunda opción teniendo el archivo *datos.dat* como el que contenga todos los ecos de los seis sonares.

Para activar los sonares el programa 1 Apéndice A se utilizó en la configuración *Individual* y el Programa 2 para la configuración *Serie*.

Para tener un solo programa y realizar la activación de los sonares y la captura de los ecos , el programa 3 se utilizó como una función que será llamada cuando se active cada sonar.

Los programas para la activación y captura de ecos quedaron de la siguiente forma: Programa 4 configuración *Individual* y Programa 5 configuración *Serie* Apéndice B.

#### 4.9 Pruebas

Se configuró la tarjeta DaqCard con Microsoft Visual C++ para crear el archivo ejecutable de los Programas 4 y 5 Apéndice B.

Se obtuvo el archivo datos.dat de un objeto colocado a 40 cm en frente del arreglo de sonares (ver figura 4.13). El programa que se utilizó fue con la configuración *Individual* (Programa 4).

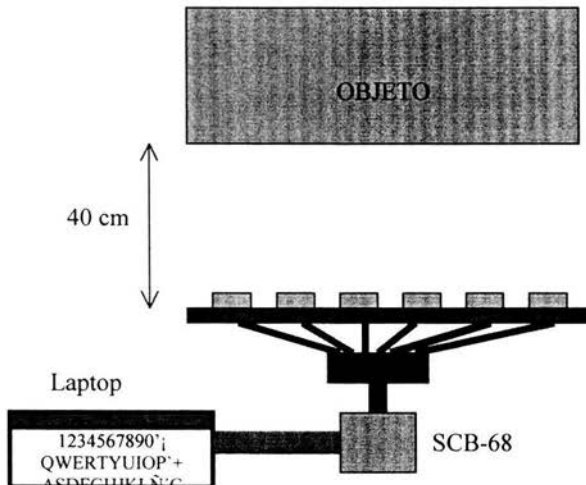


Figura 4.13 Conexión entre los dispositivos del sistema de sonares.



La visualización del archivo datos.dat se realizó bajo la plataforma MATLAB con el siguiente script:

```
load datos.dat
plot(datos(12000:15000))
hold on
plot(datos(6000:9000)+4)
plot(datos(1:3000)+8)
plot(datos(3000:6000)+12)
plot(datos(9000:12000)+16)
plot(datos(15000:18000)+20)
```

Las imágenes de los ecos se muestran en la figura.4.14

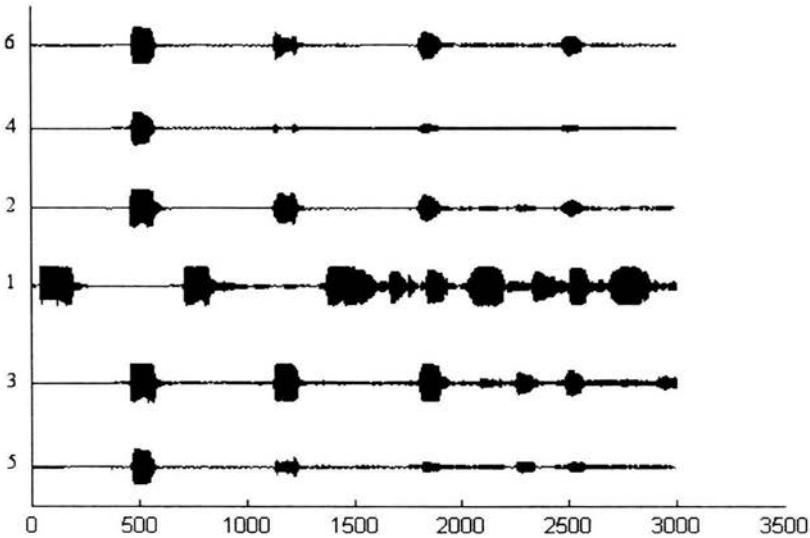


Figura 4.14 Ecos obtenidos del script

#### 4.10 Portabilidad del Programa

Microsoft Visual C++ crea archivos ejecutable con extensión *exe* de los programas 4 y 5, estos archivos ejecutables se pueden utilizar con otro código de algún programa en código C haciendo una llamada al sistema ó agregando todo el código. Para facilitar el uso de los programas se creó una cabecera contenida en el Apéndice B Programa 6, que une el programa de configuración *individual* con el de configuración *serie* .

En la secuencia individual todos los sonares transmiten y reciben, en una perspectiva es como hacer una llamada a una función donde sólo se indique el canal que se quiere que se active y obtener su eco.

Por ejemplo:

```
activar( Canal_Uno );  
activar( Canal_Dos );  
activar( Canal_Tres );
```

Para la secuencia serie sólo el sonar uno transmite y se tiene que realizar el recorrido de los sonares del uno al seis capturando los ecos. Se utilizaría entonces una llamada a una función que realice el recorrido sin necesidad de especificarle algún parámetro.

Por ejemplo:

```
escanea();
```

La cabecera tiene el nombre de **sonar**, para poder hacer uso de las funciones se tiene que declarar una variable de tipo **sonar**.

El programa en C para la secuencia individual que utilice la cabecera **sonar** es el siguiente:

```
#include <stdio.h>
#include "sonar.h"

main(){
    sonar prueba;
    prueba.chain( 1 );
}
```

En la secuencia serie se tiene el siguiente programa para hacer uso de la función `all()`.

```
#include <stdio.h>
#include "sonar.h"

main(){
    sonar prueba;
    prueba.all( );
}
```

Archivos necesarios para la compilación

```
nidaqex..h
nidaq.h
nidaqcns.h
nidaqerr.h
olderror.h
```

## CAPÍTULO V PRUEBAS Y RESULTADOS

### 5.1 Introducción

En este capítulo se explica las pruebas realizadas al sistema implementado, con el fin de ver su comportamiento y comprobar su funcionamiento. El objetivo del proyecto es implementar un sistema con un arreglo lineal de 6 sonares controlado por la computadora para capturar y visualizar las señales adquiridas. Por lo tanto las pruebas se hicieron para comprobar que el sistema funcionara y cumpliera con el objetivo principal.

### 5.2 Configuración de las pruebas.

Las pruebas se hicieron para los dos tipos de configuraciones con que cuenta el sistema, *configuración individual* y *configuración serie*.

Las pruebas se hicieron con las siguientes características.

- Computadora laptop, procesador Intel Pentium II a 600Mhz , 64Mb en RAM, sistema operativo windows 98 y ranura PCMCIA.
- Temperatura ambiente 21° C
- Para graficar los ecos reflejados se utilizó Matlab.
- Objeto rectangular de cartón (una caja), con una longitud de 80cm y una altura de 40cm.

El desarrollo de las pruebas se realizó colocando el objeto en distintos puntos del medio de transmisión así como a diferentes orientaciones, utilizando en dos tipos de configuraciones.

Se optó por un objeto en forma rectangular, (una caja) debido a que es una figura geométrica regular y uno de los trabajos a futuro es usar el sistema como un sistema de navegación para robots móviles y el objeto rectangular simula los posibles obstáculos a los que un robot se puede encontrar en su recorrido.

En la figura 5.1 se muestra el arreglo de sonares y el objeto a una distancia  $d$ . El objeto se colocó a diferentes posiciones 40cm y 70cm. La inclinación que se le dió al objeto fue  $d$ . Donde los valores de  $\alpha$  son  $0^\circ$  y  $45^\circ$ .

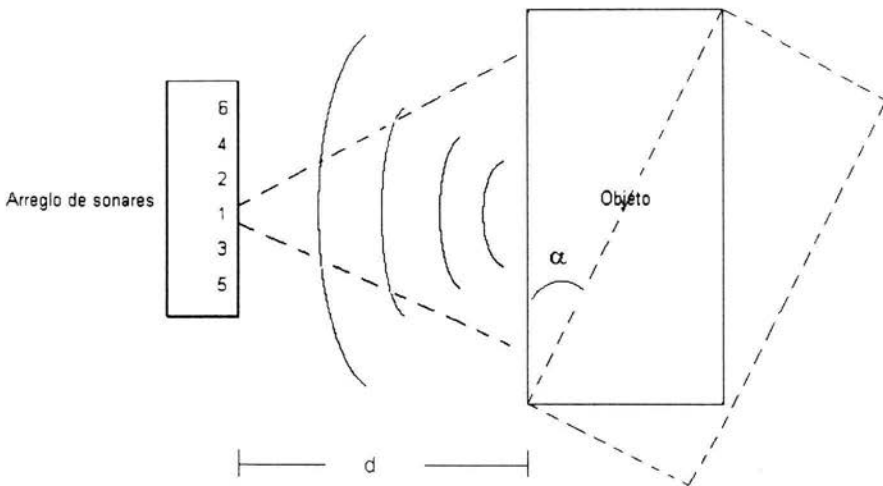


Figura 5.1 Pruebas del Sistema

## 5.2 Configuración Individual caso I.

La primera prueba que se realizó fue con la *configuración individual*.

donde:  $d = 40\text{cm}$  y  $\tilde{\alpha} = 0^\circ$

A continuación se muestra la figura 5.2 donde se pueden ver los ecos reflejados por el objeto radiado con la señal ultrasónica.

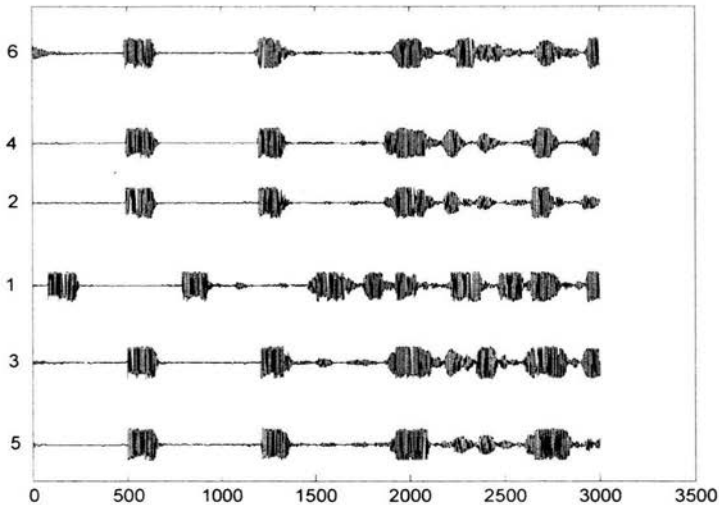


Figura 5.2 Eco reflejado caso I

Los números que aparecen en el eje Y de la figura 5.2 corresponden al número de sonar en el arreglo. Las gráficas desplegadas son visualizadas por MATLAB.

Como se puede observar en la figura 5.2 el sonar 1, quien detecta primero el eco reflejado por el objeto, debido a que este sonar fue elegido para enviar el pulso de ultrasonido y los 5 sonares restantes presentan mayor atenuación de la señal del eco reflejado por el objeto. La idea de esta configuración es obtener mayor información por los otros sonares debido al eco reflejado por el objeto, el sistema trabajo correctamente en la primer prueba.

### 5.2.1 Configuración Individual caso II.

En la segunda prueba se tienen las siguientes variables de acuerdo a la figura 5.1:

$$d = 70\text{cm y } \alpha = 0^\circ$$

en esta prueba el objeto se alejó 30cm más que en la prueba anterior, la figura 5.3 muestra las señales adquiridas a esta distancia por el sistema y visualizadas por MATLAB.

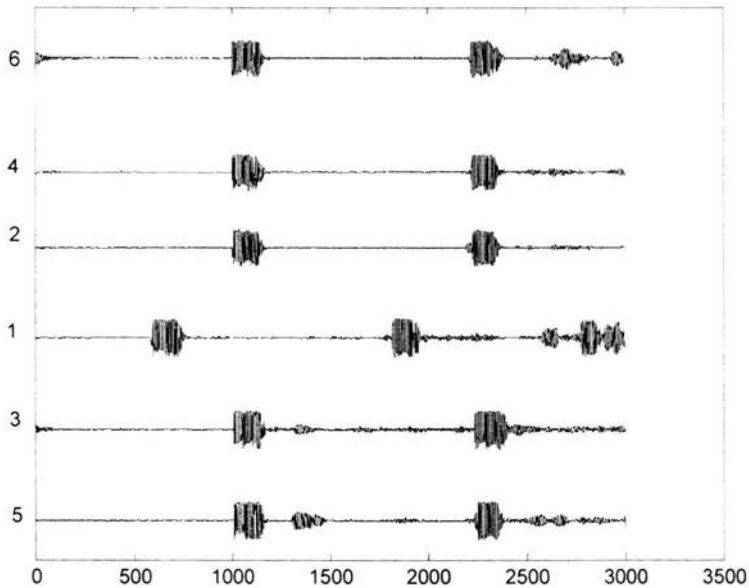


Figura 5.3 Eco reflejado caso II

El comportamiento del sistema fue tal y como se esperaba, se puede apreciar en la figura 5.3 que existe una mayor atenuación si el objeto se coloca a una distancia mayor del sistema.

El sonar 1 es el primero en detectar el eco reflejado con una mayor atenuación que en la figura 5.2, mientras los 5 sonares restantes detectan el eco reflejado al mismo tiempo pero con una mayor atenuación que el sonar 1, en general las señales presentan una mayor atenuación con respecto a las de la figura 5.2 y esto es debido a que la distancia entre el objeto y el sistema se incremento.

### 5.2.2 Configuración Individual caso III.

En la siguiente prueba realizada se continuó con la *configuración individual*, los valores en esta prueba son  $d = 60\text{cm}$  y  $\alpha = 45^\circ$ . Las señales del eco reflejado son mostradas con MATLAB y se muestran en la figura 5.4

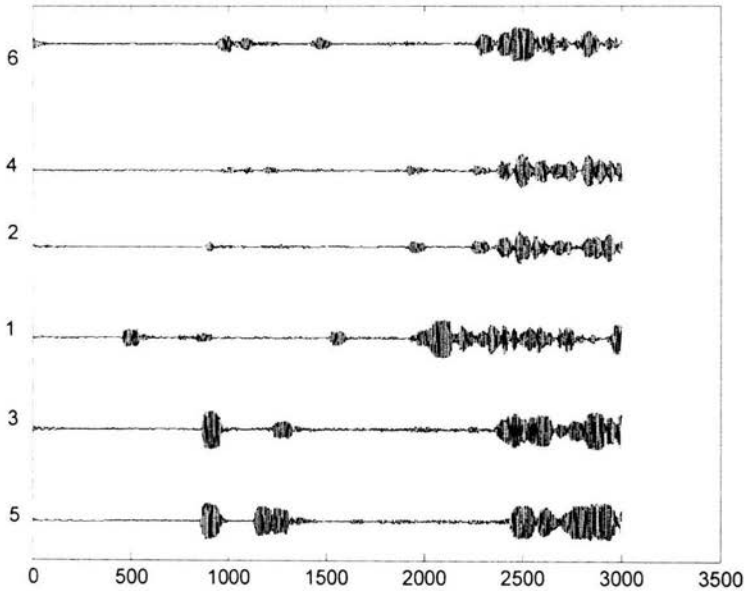


Figura 5.4 Ecos reflejados caso III

El resultado es el esperado, debido a que el objeto tiene una inclinación de  $45^\circ$ , la propagación del pulso ultrasónico no es normal a la superficie del objeto, por lo que se generan difracciones en todas direcciones provocando que el eco detectado esté muy atenuado.



### 5.3 Configuración serie caso I.

En esta prueba el sistema fue configurado en *serie*, donde  $d = 40\text{cm}$  y  $\alpha = 5^\circ$

La figura 5.5 muestra las señales recibidas.

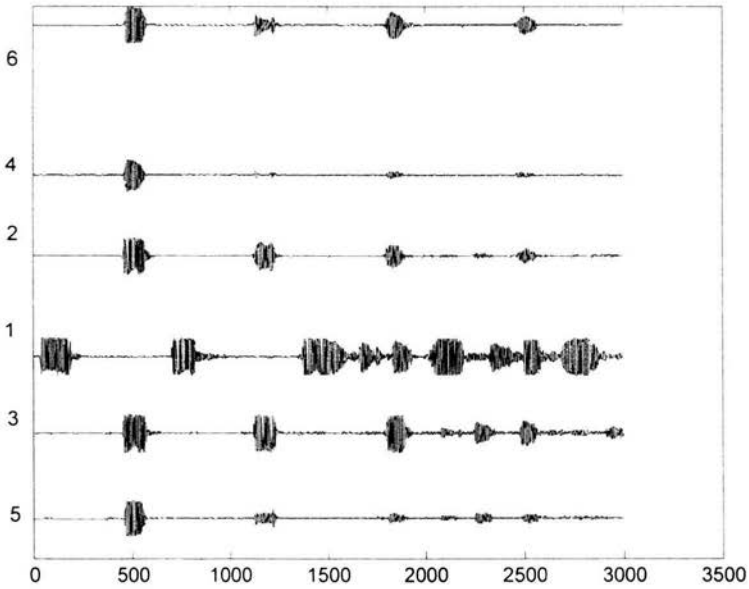


Figura 5.5. Ecos reflejados Caso I *configuración serie*

Donde la única diferencia visible es que el sonar 1 recibe el eco reflejado con menor atenuación que los otros 5 sonares restantes. La diferencia mas clara y precisa solo puede ser observada al aplicar algún algoritmo de reconstrucción del objeto.

### 5.3.2 Configuración Serie caso II

El objeto se encuentra a  $d = 70\text{cm}$  y  $\alpha = 45^\circ$

En la figura 5.6 se muestra las señales obtenidas por el sistema, una vez que los sonares enviaron la señal de ultrasonido y la señal de eco reflejado es mostrada en MATLAB.

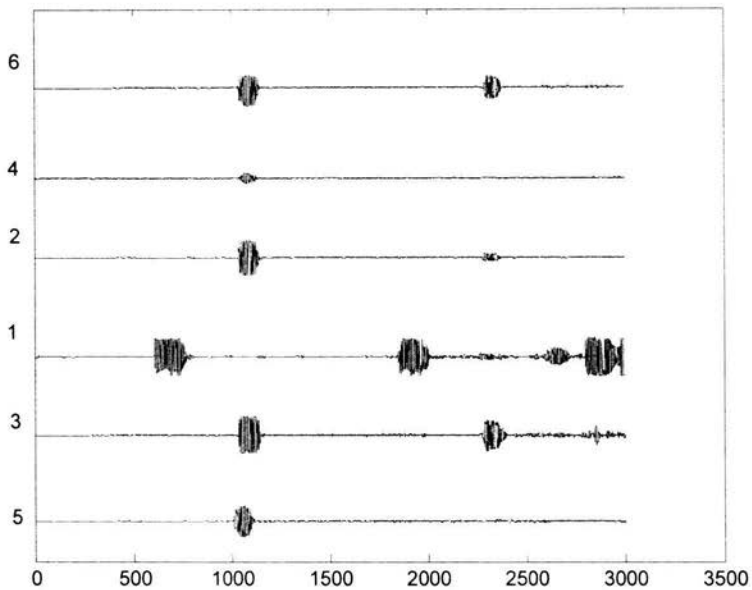


Figura 5.6 Ecos reflejados Caso II *configuración serie*

Como el objeto se colocó a una distancia mayor del sistema que en el caso I, se presenta mayor atenuación de la señal del eco reflejado. El sonar 1 es quien empieza la secuencia de transmitir y recibir, presenta una atenuación menor que los 5 sonares restantes. Los resultados son esperados ya que mientras el objeto este más alejado del sistema la atenuación tiene a ser mayor.

### 5.3.3 Configuración Serie caso III

Se sigue con la misma configuración donde  $d = 60\text{cm}$  y  $\alpha = 45^\circ$

La figura 5.7 muestra las señales obtenidas del eco reflejado por el objeto, visualizadas en MATLAB.

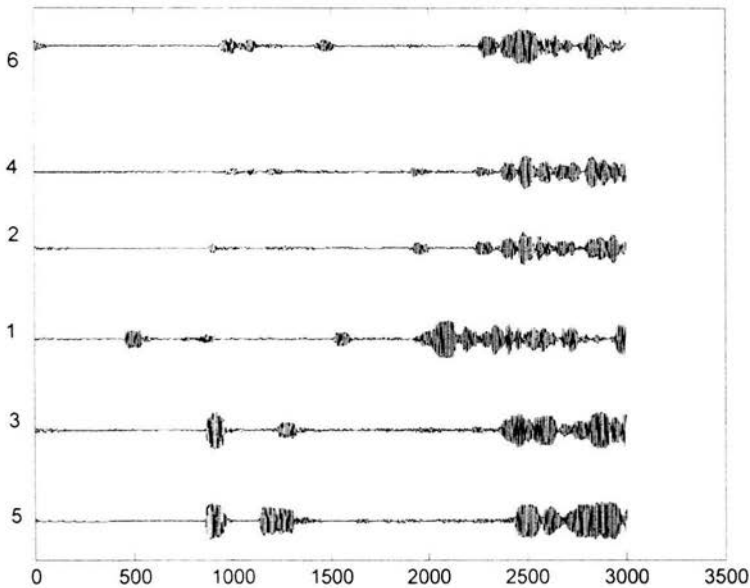


Figura 5.7 Ecos reflejados Caso III configuración serie

Los sonares 1, 3 y 5 reciben el eco reflejado por el objeto claramente, mientras que los sonares 2, 4 y 6 pierden parte de la señal. Esto es debido a la inclinación de  $45^\circ$  que se le dio al objeto ver figura 5.1, por lo tanto los sonares 2, 4 y 6 quedan más alejados del objeto que los sonares 1, 3 y 5. Puede verse en la figura 5.7 que la atenuación de la señal del eco reflejado para los sonares 2, 4 y 6 es mayor que la de los sonares 1, 3 y 5.

#### 5.4 Conclusiones.

El sistema respondió satisfactoriamente a las pruebas realizadas, en especial se observó que las señales eran más claras cuando se utilizaba la *configuración individual*, mientras que la *configuración serie* mostraba ligeras alteraciones de la señal recibida.

El motivo de por qué la *configuración individual* tuvo un mejor desempeño puede ser a que sólo un sonar es seleccionado para enviar el pulso de ultrasonido y los 5 sonares restantes son receptores, esto evita que las señales de ultrasonido que pudieran ser enviadas por los 5 sonares restantes se traslapen generando ruido entre los sonares, afectando la claridad de recepción de la señal del eco reflejado por el objeto como se, pudo observar en la *configuración serie*.

Al ver las señales visualizadas por MATLAB es difícil determinar a simple vista que configuración proporciona mayor información del objeto, si la *configuración individual* o la *configuración serie*. Es necesario procesar esas señales por medio de algoritmos de reconstrucción de señales para precisar cual configuración entrega información suficiente para reconstruir el objeto.

Las pruebas reflejan que el sistema implementado en este proyecto, funciona correctamente para cualquiera de las dos configuraciones que se quiera usar. El control del sistema por la computadora y la visualización de las señales en tiempo real, funcionan de manera óptima y puede ser usado para la reconstrucción de objetos con su debido tratamiento de las señales para ser aplicado a un sistema de navegación para un robot móvil

## CAPÍTULO VI CONCLUSIONES Y TRABAJO A FUTURO

### 6.1 Conclusiones

Durante el desarrollo de este trabajo se han generado múltiples conclusiones que son parte fundamental de los objetivos de este proyecto. A continuación se muestra un compendio de las conclusiones más importantes.

- Fue necesario modificar la tarjeta de medición POLAROID para que los sonares funcionaran como receptores únicamente. Pero sin perder la configuración original, es decir que fuera posible elegir entre los dos tipos de configuraciones: *configuración serie* y *configuración individual*.
- La tarjeta de medición POLAROID en su configuración original entrega la señal ya procesada. En los objetivos del proyecto se quería que la señal obtenida fuera sin procesar para que en el trabajo futuro se le puedan aplicar algoritmos apropiados. Por lo tanto fue necesario obtener la señal antes de ser procesada, es la razón por la cual no se tomó la señal procesada que entrega normalmente la tarjeta de medición POLAROID.
- Se pudo comprobar que en general la tarjeta de medición POLAROID es excelente para ser modificada cuando se requiere su uso en sistemas de detección de objetos experimentales.

- La tarjeta NIDAQ se acopla muy bien con los sonares POLAROID sin embargo si no se tiene cuidado en la hora de programar el control de la tarjeta con los sonares se corre el riesgo de que la frecuencia de muestreo de 256kHz se divida entre el número de sonares que se tienen en el arreglo. Si se tiene ese cuidado cuando se esta programando el control entre la tarjeta y el arreglo de sonares es posible adecuar la tarjeta NIDAQ para que la frecuencia de muestreo no se divida entre el número de sonares y se tenga una frecuencia igual a 256kHz para cada sonar, como fue el caso del proyecto.
  
- Los controladores de la tarjeta NIDAQ son de fácil manejo gracias a las librerías que están escritas en un lenguaje bastante robusto como lo es "C", esto nos dio la facilidad de realizar los cambios y adecuarlas a nuestras necesidades.
  
- La unidad SCB68 fue necesaria para tener un orden entre la conexión física de la tarjeta NIDAQ con el arreglo de sonares. Permitiendo tener un cableado estructurado y ordenado entre ambos objetos.
  
- Fue necesario codificar un archivo cabecera "sonar.h" para facilitar el desarrollo de futuras aplicaciones del sistema implementado. Al incluir "sonar.h" al inicio de cada aplicación es posible controlar los dos tipos de configuraciones (*configuración serie, configuración individual*) implementadas al sistema. Permitiendo aplicar algoritmos avanzados a las señales obtenidas por el sistema.
  
- Se encontraron problemas con el sistema operativo windows 98, el sistema se congelaba después de una hora de operación y era necesario reiniciar el sistema. En varias ocasiones fue necesario reinstalar el sistema por completo, ya que el sistema operativo afectaba el control del sistema implementado, dejándolo inútil por largos periodos de tiempo.

- Cuando se instaló en la computadora el sistema operativo windows 2000, se pudo ver que el sistema implementado trabajó mucho mejor que en windows 98. Ya no se tuvo el problema de que el sistema se congelaba y no hubo la necesidad de reiniciar el sistema, se pudo tener al sistema trabajando por más de 3 horas sin ningún error aparente.
  
- El sistema funcionó de manera correcta en un ambiente real, que es donde se hicieron todas las pruebas. Con ambiente real nos referimos a un ambiente con ruido ajeno al sistema y objetos a su alrededor. Cabe mencionar que un ambiente en el cual se aislará todo el ruido y no existieran objetos cercanos, la señal del eco reflejado por el objeto sería casi perfecta.
  
- Las pruebas realizadas muestran a simple vista que la *configuración individual*, muestra una señal mas clara del eco reflejado. Debido a que un solo sonar es quien envía el pulso de ultrasonido, evitando un traslape entre los sonares restantes si fuesen configurados para enviar un pulso de ultrasonido.
  
- Los dos tipos de configuraciones realizadas dan la posibilidad de realizar una gama amplia de pruebas, para comparar las señales adquiridas por el sistema y determinar con la ayuda de algoritmos de procesamiento de señales que configuración es la que otorga mayor información para la reconstrucción del objeto.

## **6.2 Trabajo a futuro y recomendaciones.**

El sistema implementado pretende ser un punto de partida para nuevas investigaciones y desarrollos sobre el área de procesamiento de señales y navegación en robots móviles. La siguiente lista presenta algunos posibles caminos para mejoras del proyecto o el desarrollo de nuevas investigaciones a partir de él.

- El sistema puede ser usado para investigaciones sobre nuevos algoritmos de reconstrucción de objetos.
  
- Empleo de algoritmos propios del tratamiento de señales, para realizar las pruebas pertinentes y determinar qué configuración resulta más eficiente para reconstruir el objeto.
  
- El proyecto implementado puede ser usado como sistema de navegación para un robot móvil.
  
- Se promueve al desarrollo de controladores de la tarjeta NIDAQ para sistemas UNIX o LINUX.
  
- Desarrollar un sistema de rotación automática que pueda ser adaptado al sistema desarrollado. Con el objetivo que el sistema tenga una mejor visión de su entorno.
  
- Adaptar el sistema desarrollado a cualquier aplicación donde se requiera la navegación por sonares.
  
- Desarrollar una interfaz gráfica y amigable que permita controlar el sistema.
  
- Eliminar el switch manual que permite la selección de sonares, desarrollando un switch automático con algún dispositivo lógico programable.



## Apéndice A

### Programa 1

```
/*  
*  
* Información de Conexión:  
* La señal digital se obtiene en el puerto 0, en la línea 0. Conectando el led a tierra en el pin DIG GND.  
*  
***/  
  
#include "nidaqex.h"  
void main(void){  
    /*  
    * Declaración de Variables Locales: */  
    i16 iStatus = 0;  
    i16 iRetVal = 0;  
    i16 iDevice = 1;  
    i16 iPort = 0;  
    i16 iLine = 0;  
    i16 iDir = 1;  
    i16 iStateON = 1;  
    i16 iStateOFF = 0;  
    i16 iIgnoreWarning = 0;  
    /* Configurando linea de salida. */  
    iStatus = DIG_Line_Config(iDevice, iPort, iLine, iDir);  
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Line_Config", iIgnoreWarning);  
    /* Enviando el pulso de salida. */  
    iStatus = DIG_Out_Line(iDevice, iPort, iLine, iStateON);  
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Line", iIgnoreWarning);  
}
```

---

## Programa 2

```

/*****
* Descripción:
*  Obtiene una señal del canal análogo de entrada
* Conexiones en los pines:
*  Conectando la señal análoga al canal AI 1.
*****/

#include "nidaqex.h"
void main(void){
    // Declaración de variables locales:
    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iDevice = 1;
    i16 iChan = 1;
    i16 iGain = 1;
    f64 dSampRate = 1000.0;
    u32 ulCount = 100;
    f64 dGainAdjust = 1.0;
    f64 dOffset = 0.0;
    i16 iUnits = 0;
    i16 iSampTB = 0;
    u16 uSampInt = 0;
    static i16 piBuffer[100] = {0};
    i16 iDAQstopped = 0;
    u32 ulRetrieved = 0;
    static f64 pdVoltBuffer[100] = {0.0};
    i16 ignoreWarning = 0;
    i16 iYieldON = 1;

    iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt);
    /* Adquisición de datos para un canal*/
    iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt);
    iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Start", ignoreWarning);
    while ((iDAQstopped != 1) && (iStatus == 0)) {

```

```
/* Ciclo hasta que la adquisición este completada. */
    iStatus = DAQ_Check(iDevice, &iDAQstopped, &ulRetrieved);
    iRetVal = NIDAQYield(iYieldON);
}
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Check", iIgnoreWarning);
iStatus = DAQ_VScale(iDevice, iChan, iGain, dGainAdjust, dOffset, ulCount, piBuffer, pdVoltBuffer);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_VScale", iIgnoreWarning);
iStatus = DAQ_Clear(iDevice);
/* Plot acquired data */
iRetVal = NIDAQPlotWaveform(pdVoltBuffer, ulCount, WFM_DATA_F64);
printf(" The data is available in 'pdVoltBuffer'.\n");
}
```

## APÉNDICE B

### Programa 1

```
/*Programa que activa los sonares en configuración individual*/
```

```
#include "nidaqex.h"

void main(void){
    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iDevice = 1;
    i16 iPort = 0;
    i16 iMode = 0;
    i16 iDir = 1;
    i32 iPattern = 1;
    i32 iNumLoops = 1;
    i16 iIgnoreWarning = 0;
    i16 iYieldON = 1;
    int contador;
    int serie [7];
    serie[1]=1;           //000001
    serie[2]=2;           //000010
    serie[3]=4;           //000100
    serie[4]=8;           //001000
    serie[5]=16;          //010000
    serie[6]=32;          //100000

    for(contador=1; contador!=7; ++contador){
        iNumLoops=serie[contador];
        iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
        iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);
        while ((iPattern < iNumLoops) && (iStatus == 0)) {
            iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
            iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", iIgnoreWarning);
            /*printf(" Salida digital en puerto %d en (DECIMAL) %ld\n", iPort, iPattern);*/
            iRetVal = NIDAQDelay(0.0);
        }
    }
}
```

---

```
        iPattern=serie[contador];
        iRetVal = NIDAQYield(iYieldON);

    }
}
}
```

## Programa 2

/\* Programa que activa los sonares en configuración serie \*/

```
#include "nidaqex.h"
void main(void){
    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iDevice = 1;
    i16 iPort = 0;
    i16 iMode = 0;
    i16 iDir = 1;
    i32 iPattern = 1;
    i32 iNumLoops = 1;
    i16 iIgnoreWarning = 0;
    i16 iYieldON = 1;
    int contador;
    int serie [7];
    serie[1]=1;           //000001
    serie[2]=3;           //000011
    serie[3]=5;           //000101
    serie[4]=9;           //001001
    serie[5]=17;          //010001
    serie[6]=33;          //100001

    for(contador=1; contador!=7; ++contador){
        iNumLoops=serie[contador];
        iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
        iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);
```

---

```

while ((iPattern < iNumLoops) && (iStatus == 0)) {
    iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", iIgnoreWarning);
    /*printf(" Salida digital en puerto %d en (DECIMAL) %ld\n", iPort, iPattern);*/
    iRetVal = NIDAQDelay(0.0);
    iPattern=serie[contador];
    iRetVal = NIDAQYield(iYieldON);

}
}
}

```

### Programa 3

```

/*Programa que captura los ecos del cana 1 */

```

```

#include "nidaqex.h"
void main(void){
    i16 iStatus = 0;
    i16 iChan = 1;
    i16 iRetVal = 0;
    i16 iGain = 1;
    i16 iDevice = 1;
    f64 dSampRate = 250000.0;
    u32 ulCount = 3000;
    f64 dGainAdjust = 1.0;
    f64 dOffset = 0.0;
    i16 iUnits = 0;
    i16 iSampTB = 0;
    u16 uSampInt = 0;
    static i16 piBuffer[3000] = {0};
    i16 iDAQstopped = 0;
    u32 ulRetrieved = 0;
    static f64 pdVoltBuffer[3000] = {0.0};
    i16 iIgnoreWarning = 0;
    i16 iYieldON = 1;

```

---

```

FILE *ptrstream;
int j;

iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt);
iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Start", iIgnoreWarning);
while ((iDAQstopped != 1) && (iStatus == 0)) {
    iStatus = DAQ_Check(iDevice, &iDAQstopped, &ulRetrieved);
    iRetVal = NIDAQYield(iYieldON);
}
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Check", iIgnoreWarning);
iStatus = DAQ_VScale(iDevice, iChan, iGain, dGainAdjust, dOffset, ulCount, piBuffer, pdVoltBuffer);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_VScale", iIgnoreWarning);
iStatus = DAQ_Clear(iDevice);
ptrstream=fopen("datos.dat","a");
for (j=0; j<ulCount; j++) {
    /*printf("%f\n", pdVoltBuffer[j]);*/
    fprintf(ptrstream,"%f\n", pdVoltBuffer[j]);
}
fclose(ptrstream);
}

```

#### Programa 4

*/\* Programa que active y captura los ecos en configuración individual \*/*

```

#include "nidaqex.h"
i16 iChan = 1;
void main(void)
{
    int scaneo(int iChan);
    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iDevice = 1;
    i16 iPort = 0;
    i16 iMode = 0;

```

---

```
i16 iDir = 1;
i32 iPattern = 1;
i32 iNumLoops = 1;
i16 iIgnoreWarning = 0;
i16 iYieldON = 1;

int contador;
int serie [7];
serie[1]=1;           //000001
serie[2]=2;           //000010
serie[3]=4;           //000100
serie[4]=8;           //001000
serie[5]=16;          //010000
serie[6]=32;          //100000

for(contador=1; contador!=7; ++contador){
    iNumLoops=serie[contador];
    iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);
    while ((iPattern < iNumLoops) && (iStatus == 0)) {
        iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
        iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", iIgnoreWarning);
        /*printf(" Salida digital en puerto %d en (DECIMAL) %ld\n", iPort, iPattern);*/
        iRetVal = NIDAQDelay(0.0);
        iPattern=serie[contador];
        iRetVal = NIDAQYield(iYieldON);
        if ((iPattern%2)==0){
            scaneo(iChan);
            ++iChan;
        }
    }
}
}
```

---



---

```

scaneo (iChan){

    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iGain = 1;
    i16 iDevice = 1;
    f64 dSampRate = 250000.0;
    u32 ulCount = 3000;
    f64 dGainAdjust = 1.0;
    f64 dOffset = 0.0;
    i16 iUnits = 0;
    i16 iSampTB = 0;
    u16 uSampInt = 0;
    static i16 piBuffer[3000] = {0};
    i16 iDAQstopped = 0;
    u32 ulRetrieved = 0;
    static f64 pdVoltBuffer[3000] = {0.0};
    i16 ilgnoreWarning = 0;
    i16 iYieldON = 1;
    FILE *ptrstream;
    int j;

    iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt);
    iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt);
    iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Start", ilgnoreWarning);
    while ((iDAQstopped != 1) && (iStatus == 0)) {
        iStatus = DAQ_Check(iDevice, &iDAQstopped, &ulRetrieved);
        iRetVal = NIDAQYield(iYieldON);
    }
    iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Check", ilgnoreWarning);
    iStatus = DAQ_VScale(iDevice, iChan, iGain, dGainAdjust, dOffset, ulCount, piBuffer, pdVoltBuffer);
    iRetVal = NIDAQErrorHandler(iStatus, "DAQ_VScale", ilgnoreWarning);
    iStatus = DAQ_Clear(iDevice);
    ptrstream=fopen("datos.dat","a");
    for (j=0; j<ulCount; j++) {
        /*printf("%f\n", pdVoltBuffer[j]);*/
        fprintf(ptrstream,"%f\n", pdVoltBuffer[j]);
    }
}

```

---

```
}  
fclose(ptrstream);  
return;  
}
```

## Programa 5

*/\* Programa que active y capture los ecos en configuración serie \*/*

```
#include "nidaqex.h"  
i16 iChan = 1;  
  
void main(void){  
int scaneo(int iChan);  
i16 iStatus = 0;  
i16 iRetVal = 0;  
i16 iDevice = 1;  
i16 iPort = 0;  
i16 iMode = 0;  
i16 iDir = 1;  
i32 iPattern = 1;  
i32 iNumLoops = 1;  
i16 iIgnoreWarning = 0;  
i16 iYieldON = 1;  
int contador;  
int serie [7];  
serie[1]=1;           //000001  
serie[2]=3;           //000011  
serie[3]=5;           //000101  
serie[4]=9;           //001001  
serie[5]=17;          //010001  
serie[6]=33;          //100001  
  
for(contador=1; contador!=7; ++contador){  
iNumLoops=serie[contador];
```

---

```

iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", ignoreWarning);
while ((iPattern < iNumLoops) && (iStatus == 0)) {
    iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", ignoreWarning);
    /*printf(" Salida digital en puerto %d en (DECIMAL) %ld\n", iPort, iPattern);*/
    iRetVal = NIDAQDelay(0.0);
    iPattern=serie[contador];
    iRetVal = NIDAQYield(iYieldON);
    if ((iPattern%2)==0){
        scaneo(iChan);
        ++iChan;
    }
}
}
}
scaneo (iChan){

    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iGain = 1;
    i16 iDevice = 1;
    f64 dSampRate = 250000.0;
    u32 ulCount = 3000;
    f64 dGainAdjust = 1.0;
    f64 dOffset = 0.0;
    i16 iUnits = 0;
    i16 iSampTB = 0;
    u16 uSampInt = 0;
    static i16 piBuffer[3000] = {0};
    i16 iDAQstopped = 0;
    u32 ulRetrieved = 0;
    static f64 pdVoltBuffer[3000] = {0.0};
    i16 ignoreWarning = 0;
    i16 iYieldON = 1;
    FILE *ptrstream;
    int j;

```

---

```
iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt);
iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Start", iIgnoreWarning);
while ((iDAQstopped != 1) && (iStatus == 0)) {
    iStatus = DAQ_Check(iDevice, &iDAQstopped, &ulRetrieved);
    iRetVal = NIDAQYield(iYieldON);
}
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Check", iIgnoreWarning);
iStatus = DAQ_VScale(iDevice, iChan, iGain, dGainAdjust, dOffset, ulCount, piBuffer, pdVoltBuffer);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_VScale", iIgnoreWarning);
iStatus = DAQ_Clear(iDevice);
ptrstream=fopen("datos.dat","a");
for (j=0; j<ulCount; j++) {
    /*printf("%f\n", pdVoltBuffer[j]);*/
    fprintf(ptrstream,"%f\n", pdVoltBuffer[j]);
}
fclose(ptrstream);
return;
}
```

## Programa 6

```
#ifndef sonar_h
#define sonar_h
#include "nidaqex.h"

class sonar{

public:
    void all();
    void chain(int canal);

private:
    i16 iChan = 1;
    int scaneo(int iChan, int opc);
}
```

---

```
        i16 iStatus = 0;
        i16 iRetVal = 0;
        i16 iDevice = 1;
        i16 iPort = 0;
        i16 iMode = 0;
        i16 iDir = 1;
        i32 iPattern = 1;
        i32 iNumLoops = 1;
        i16 iIgnoreWarning = 0;
        i16 iYieldON = 1;
};

sonar::sonar(){ }
sonar::~~sonar(){ }

////////////////////////////////////E S C A N E O////////////////////////////////////

void sonar::scaneo(int iChain, int opc){
    i16 iStatus = 0;
    i16 iRetVal = 0;
    i16 iGain = 1;
    i16 iDevice = 1;
    f64 dSampRate = 250000.0;
    u32 ulCount = 3000;
    f64 dGainAdjust = 1.0;
    f64 dOffset = 0.0;
    i16 iUnits = 0;
    i16 iSampTB = 0;
    u16 uSampInt = 0;
    static i16 piBuffer[3000] = {0};
    i16 iDAQstopped = 0;
    u32 ulRetrieved = 0;
    static f64 pdVoltBuffer[3000] = {0.0};
    i16 iIgnoreWarning = 0;
    i16 iYieldON = 1;
    FILE *ptrstream;
    int j;
```

---

---

```

iStatus = DAQ_Rate(dSampRate, iUnits, &iSampTB, &uSampInt);
iStatus = DAQ_Start(iDevice, iChan, iGain, piBuffer, ulCount, iSampTB, uSampInt);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Start", iIgnoreWarning);

while ((iDAQstopped != 1) && (iStatus == 0)) {
    iStatus = DAQ_Check(iDevice, &iDAQstopped, &ulRetrieved);
    iRetVal = NIDAQYield(iYieldON);
}

iRetVal = NIDAQErrorHandler(iStatus, "DAQ_Check", iIgnoreWarning);
iStatus = DAQ_VScale(iDevice, iChan, iGain, dGainAdjust, dOffset, ulCount, piBuffer, pdVoltBuffer);
iRetVal = NIDAQErrorHandler(iStatus, "DAQ_VScale", iIgnoreWarning);
iStatus = DAQ_Clear(iDevice);

if ( opc == 0 )
    ptrstream=fopen("datos.dat","a");
else
    ptrstream=fopen("datos.dat","w");

for (j=0; j<ulCount; j++) {
    fprintf(ptrstream,"%f\n", pdVoltBuffer[j]);
}

fclose(ptrstream);
return;
}

////////////////////////////////////T O D O////////////////////////////////////

void sonar::all(){
    int contador;
    int serie [16];
    serie[1]=1;
    serie[2]=4;
    serie[3]=7;
    serie[4]=8;

```

---

```

serie[5]=13;
serie[6]=16;
serie[7]=25;
serie[8]=32;
serie[9]=49;
serie[10]=64;
serie[11]=97;
serie[12]=128;
serie[13]=128;
serie[14]=128;
serie[15]=128;

for(contador=1; contador!=16; ++contador){
    iNumLoops=serie[contador];
    iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", iIgnoreWarning);

    while ((iPattern < iNumLoops) && (iStatus == 0)) {
        iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
        iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", iIgnoreWarning);
        iRetVal = NIDAQDelay(0.0);
        iPattern=serie[contador];
        iRetVal = NIDAQYield(iYieldON);

        if ((iPattern%2)==0){
            scaneo(iChan, 0);
            ++iChan;
        }
    }
}
}
}

```

/////////////////////////////////C A N A L////////////////////////////////////

---

```
void sonar::chain(int canal){
    int contador=1;
    int serie [3];
    serie[1]=64;
    serie[2]=128;
    serie[3]=128;
    iChan=canal;
    iPattern=canal;

    iNumLoops=serie[1];
    iStatus = DIG_Prt_Config(iDevice, iPort, iMode, iDir);
    iRetVal = NIDAQErrorHandler(iStatus, "DIG_Prt_Config", ignoreWarning);

    while ((iPattern < iNumLoops) && (iStatus == 0)) {
        iStatus = DIG_Out_Prt(iDevice, iPort, iPattern);
        iRetVal = NIDAQErrorHandler(iStatus, "DIG_Out_Prt", ignoreWarning);
        iRetVal = NIDAQDelay(0.0);
        iRetVal = NIDAQYield(iYieldON);

        if (iPattern==canal){
            scaneo(iChan, 1);
        }

        iPattern=iNumLoops;;
        iNumLoops=serie[contador+1];
    }
    iStatus = DIG_Out_Prt(iDevice, iPort, 128);
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#endif
```

---



## BIBLIOGRAFÍA

- [1] <http://www.robotics.com/robot/sonar.html>
- [2] <http://www.cs.umd.edu/users/musliner/sonar/notes>
- [3] <http://www.acroname.com/robotics/parts/R11-6500.html>
- [4] “Ultrasonic Ranging System 6500 HandBook” Polaroid Corporation.
- [5] “NI-DAQ for PC version 6.8 User Manual” National Instruments <http://www.ni.com>
- [6] “DAQ Hardware Overview Guide” National Instruments <http://www.ni.com>
- [7] Gerald D. Maslin, “A simple Ultrasonic Ranging System”, 102nd Convention of the Audio Engineering Society at Cincinnati Ohio, 1983
- [8] Kinsler L. E., Frey A. R., “Fundamentals of Acoustics”, John Wiley and Sons Inc., New York, 1962.
- [9] Kino G. S., “Acoustic Waves: Devices, Imaging and Analog Signal Processing”, Prentice-Hall, New Jersey, 1987.
- [10] Borenstein J., Koren Y., “Obstacle Avoidance with Ultrasonic Sensors”, IEEE J. Robot. Autom. 4, pp. 213-218, 1988.
- [11] Brown M. K., “Feature extraction techniques for recognizing solid objects with an ultrasonic range sensor”, IEEE Trans. Robot. Automat. 1, pp. 191-205, 1985.

- [12] *C How to Program* Fourth Edition © 2004 ISBN: 0131426443
- [13] *Programación en C* Byron S. Gottfried. Ed Mc Graw Hill. Serie Schaum